

实验一 集成开发环境（IDE）

一、 问题描述

1. 实验目的：

掌握集成开发环境的基本功能

掌握程序设计的基本过程空间构造C表。

2. 实验内容：

针对自己选定的一款C++开发工具进行相关实验：

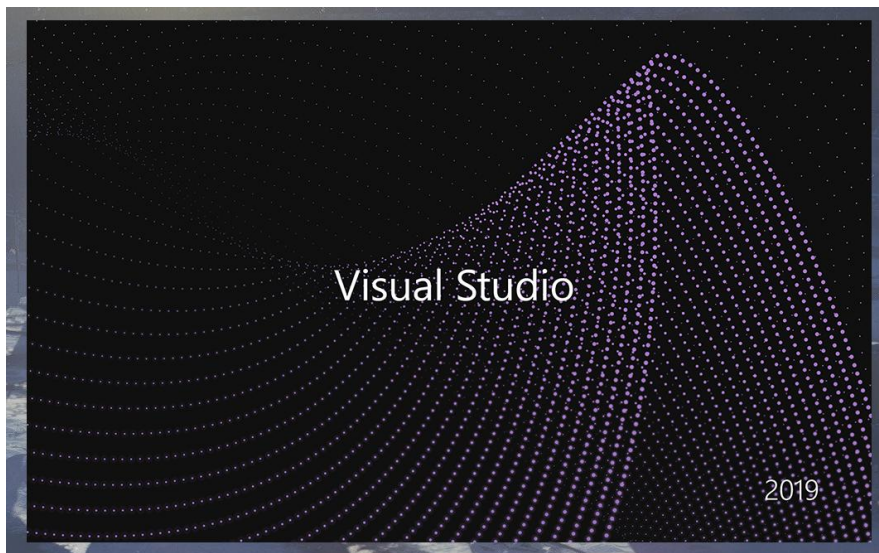
- 1、掌握项目（project）的组成；（一个常见的C++源代码包含哪些文件？各种类型的文件的用途？）
- 2、掌握编译的过程；（头文件的设置、链接库的设置、编译的选项等）
- 3、掌握调试的过程。（如何进行动态调试？设置断点、查看当前变量值等）

二、 具体实验

1. IDE选择与下载

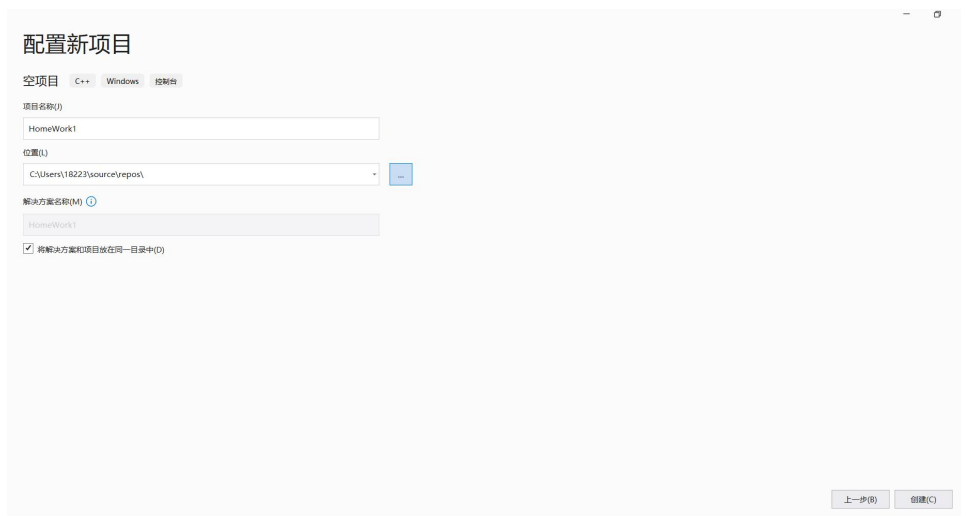
（1）选择IDE为Visual Studio 2019

（2）下载完成



2. 项目的创建

启动Visual Studio，单击创建新项目，并配置新项目。



3. 头文件设置实验

- (1) 建立Hello World项目，用来进行头文件实验。
- (2) 项目包含Main.cpp，log.cpp与log.h文件，该项目的功能是可以向控制台打印语句。
- (3) 其中Main.cpp调用InitLog()与Log()函数，向控制台打印“Hello World”，由于这两个函数并非定义在Main.cpp文件中，而是定义在了log.cpp文件中，需要在Main.cpp中对这两个函数进行声明。

这里，另外创建了log.h头文件，储存对函数的声明，于是在Main.cpp中引用log.h，即可正确运行程序。

实验截图如下：

- (1) 项目目录



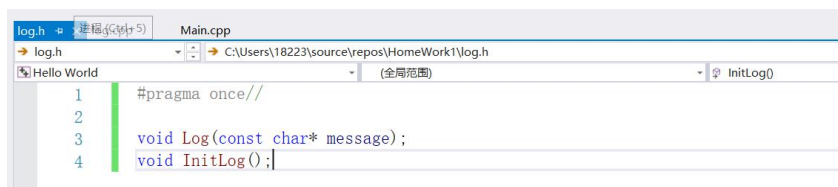
(2) Main.cpp文件



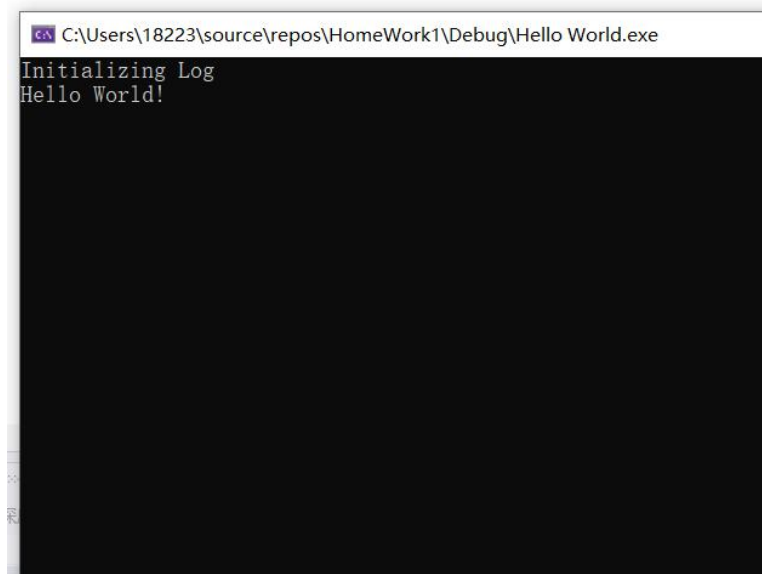
(3) log.cpp文件



(4) log.h文件



(5) 运行结果



4. 链接库的设定实验（以GLFW库为例）

（1）静态链接

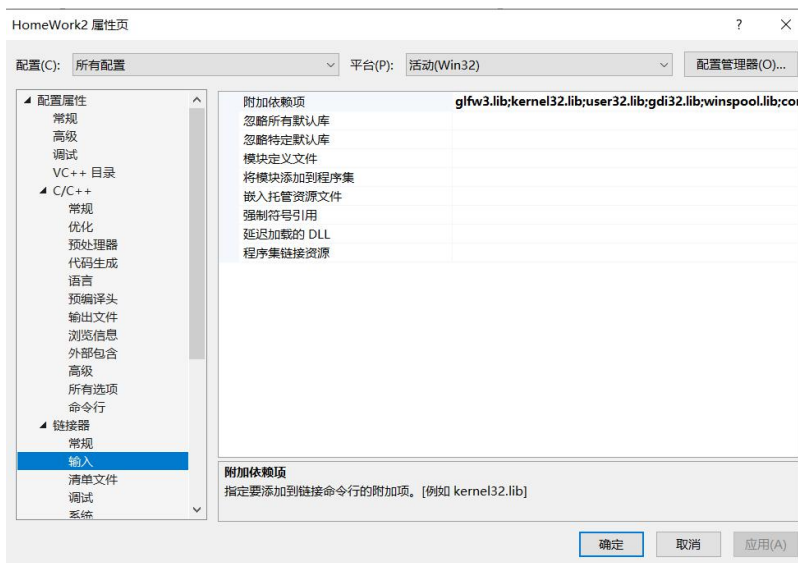
1、下载GLFW预编译二进制文件

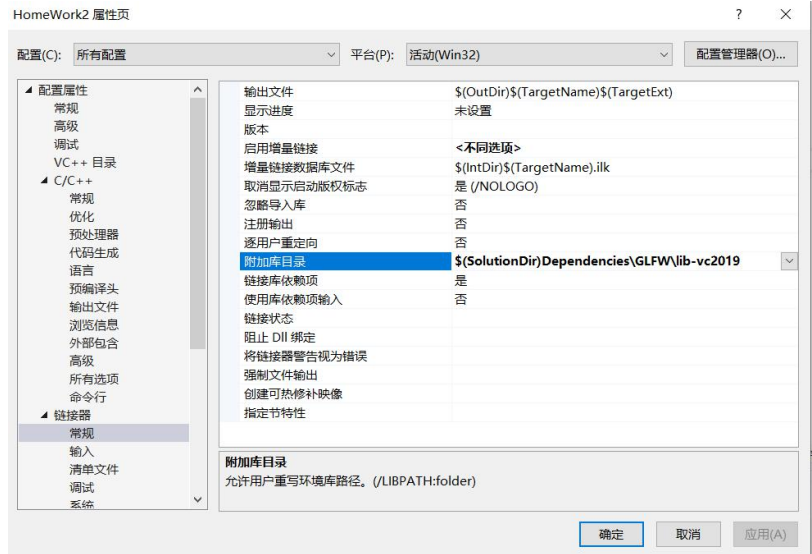
2、文件拷贝

在解决方案目录下，创建一个名为dependencies的文件夹作为库文件的目录。在文件夹内创建一个名为GLFW的文件夹，将实际下载文件放在其中。

（注：include中包含头文件，lib中包含库文件，对应2019年版本的VS，lib包含三个文件，dll是一种运行时动态链接库，glfw3dll.lib实际是一种静态库，与glfw3.dll是一起用的，这样就不需要实际询问dll。glfw3.lib即静态链接库）



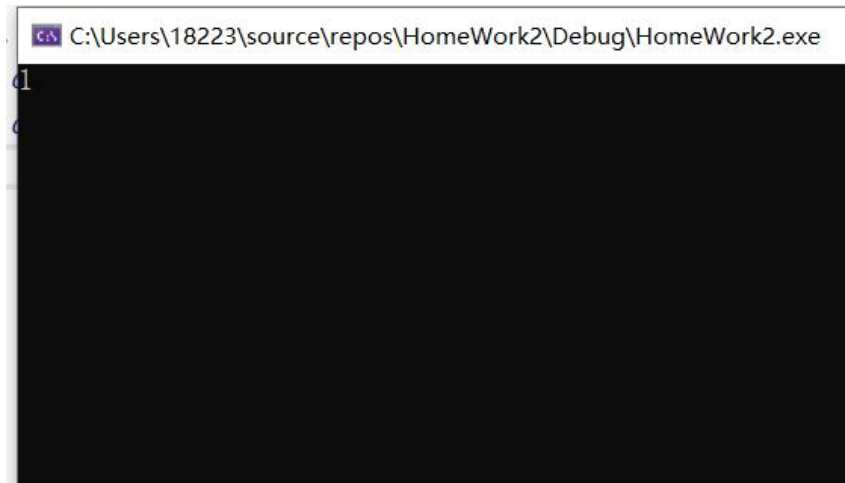




4、运行程序

在Main.cpp中引用glfw3.h文件，在main函数中调用库函数glfwInit。得到运行结果。

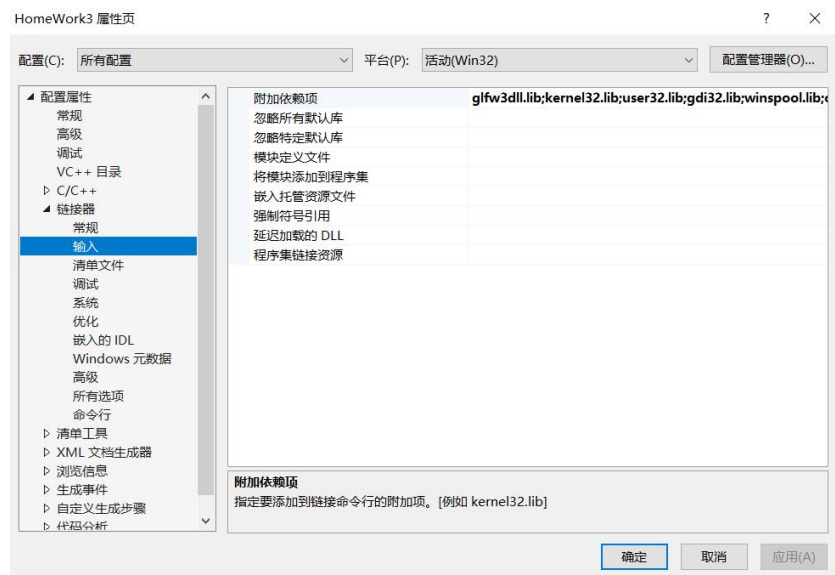
```
Main.cpp  X
Main.cpp  C:\Users\18223\source\repos\HomeWork2\Main.cpp
HomeWork2 (全局范围)
1  #include<iostream>
2  #include"glfw/glfw3.h"
3
4  int main() {
5      int a = glfwInit();
6      std::cout << a << std::endl;
7      std::cin.get();
8  }
```



(2) 动态链接

1、链接到实际的库

将之前链接的静态库用动态版本替换。



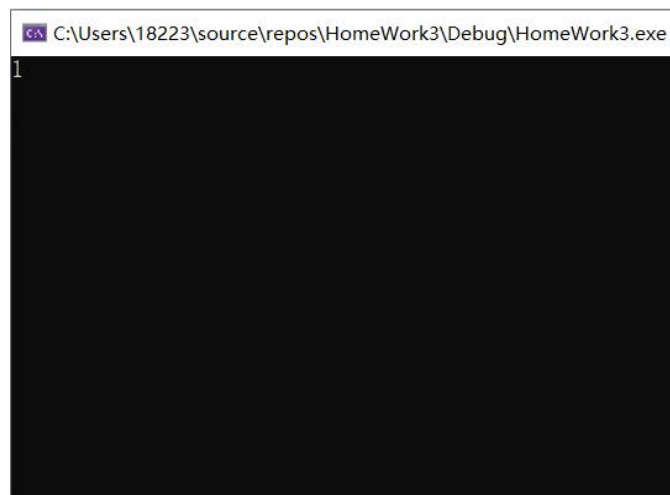
2、确保在可访问处有dll文件

将dll文件与可执行文件放在同一文件夹。

223 > source > repos > HomeWork3 > Debug

名称	修改日期	类型	大小
HomeWork3.tlog	2022/2/24 21:50	文件夹	
glfw3.dll	2022/2/24 20:46	应用程序扩展	191 KB
HomeWork3.exe	2022/2/24 21:50	应用程序	41 KB
HomeWork3.exe.recipe	2022/2/24 21:50	RECIPE 文件	1 KB
HomeWork3.ilc	2022/2/24 21:50	Incremental Link...	406 KB
HomeWork3.log	2022/2/24 21:50	文本文档	1 KB
HomeWork3.pdb	2022/2/24 21:50	Program Debug ...	1,036 KB
HomeWork3.vcxproj.FileListAbsolut...	2022/2/24 21:50	文本文档	1 KB
Main.obj	2022/2/24 21:50	3D Object	38 KB
vc142.idb	2022/2/24 21:50	VC++ Minimum ...	163 KB
vc142.pdb	2022/2/24 21:50	Program Debug ...	404 KB

3、运行程序

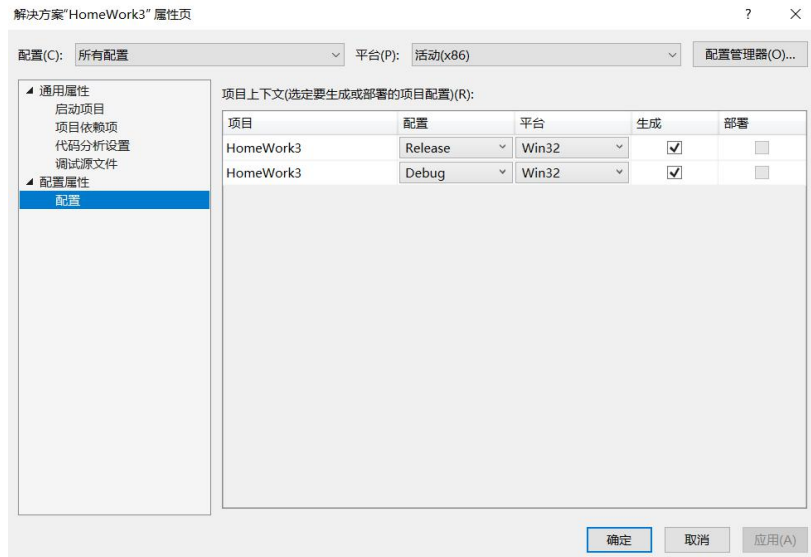


5. 编译选项

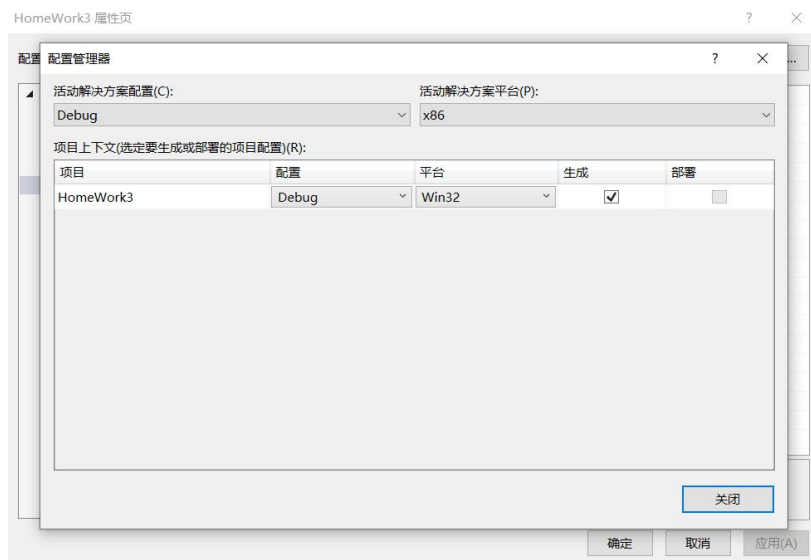
1、在解决方案属性页中设置编译选项

可通过配置管理器按钮 来维护配置集与平台集。

(注：直接通过最上面的上拉框无法改变项目设置，更改设定会影响生成dll与exe的目录。)



2、在项目中设置编译选项



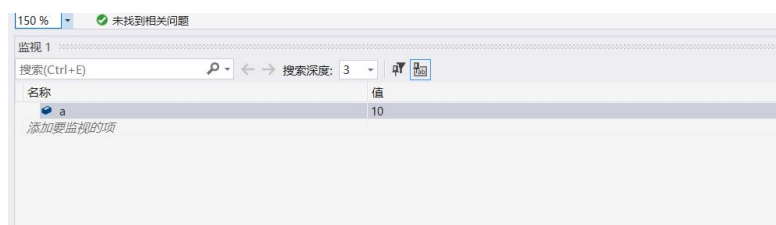
6. 动态调试实验

1、设置断点



```
1 #include <iostream>
2 #include "log.h"
3
4 int main()
5 {
6     int a = 10;
7     InitLog();
8     Log("Hello World!");
9     std::cin.get();
10    return 0;
11 }
```

2、查看当前变量值



三、 问题思考

1、掌握项目（project）的组成；（一个常见的C++源代码包含哪些文件？各种类型的文件的用途？）

答：通常一个C++程序包含如下文件：.cpp 文件和 .h 文件。其中，.cpp 文件被称作 C++ 源文件；而 .h 文件则被称作 C++ 头文件。用途如下：

- ① 头文件，一般包含类、结构、宏等的定义，一般来说，.h为扩展名的文件和.cpp 为扩展名的文件是一一对应配合使用的；
- ② cpp: 源代码文件，也叫实现文件，用来存放类的实现代码，按C++语法编译处理。

2、掌握编译的过程；(头文件的设置、链接库的设置、编译的选项等)

答：分别进行了头文件的设置，链接库的设置，编译的选项三个实验。实验内容可见（二）具体实验。

- ① 头文件的设置实验代码可见src/HomeWork1。
- ② 链接实验以GLFW库为例。链接库有静态链接和动态链接。其中静态链接在编译时发生。将库中的二进制代码和我们的程序的二进制代码合并到一个新的可执行文件中。静态链接代码见src/HomeWork2。

③ 动态链接发生在运行时期，以共享的dll文件为基础，链接时库中的代码可以不加入到可执行文件中。多个应用程序可以共享一个dll文件中的二进制代码。动态链接最后需要将dll文件放在可访问的位置（方便起见，将其与可执行文件放在一起）动态链接实验代码见**src/HomeWork3**。

④ 影响编译结果的参数有两个：编译优化和目标平台。可以在解决方案和项目设置中设置编译选项。具体内容见（二）**具体实验**。

3、掌握调试的过程。（如何进行动态调试？设置断点、查看当前变量值等）

答：见（二）**具体实验**。

四、 附录

源程序文件清单： `stdio.h` `log.h` `iostream.h`

`Main.cpp` `log.cpp` `log.h`