**操作系统实验1报告**

**原理说明：**

本实验的环境是在Windows上通过Ubuntu的开发环境模拟linux的系统，通过QEMU启动一个支持multiboot启动协议的内核，在内核中编写需要完成的输出。

**源代码说明：**

Makefile文件中分别注明了要创建的文件，即O文件与bin文件、创建文件依赖的文件列表，即ld文件，以及通过依赖文件创建目标的命令组，clean中的内容表示重新编译时会重新生成文件。

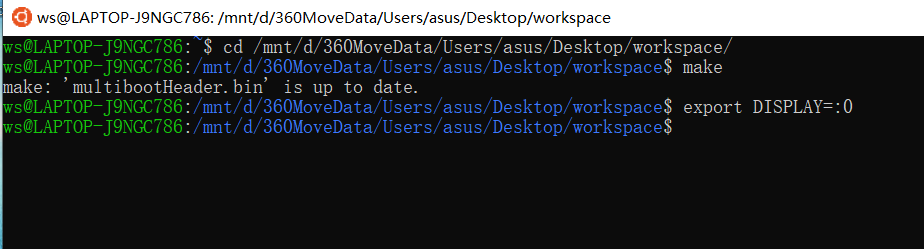
Ld文件中第一行注明了输出格式，第二行注明支持模式是x86的，第三行则表示将以汇编文件中的start为入口，Sections部分表示了代码排布，其中当前位置为1M的物理内存开始，然后代码段中以multiboot\_header开头，说明在S文件中有multiboot\_header一段代码，然后align说明是8个字节对齐，最后将VGA输出的内容置于后面。

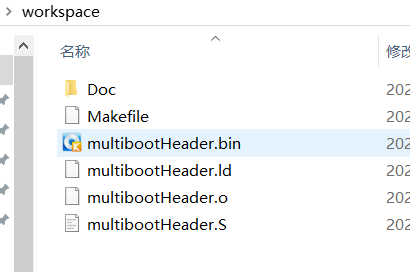
S文件说，先声明start、\_start的全局性，然后在start中写入需要在VGA上显示的内容，其中movl就是在显存上写入需要显示的数据，每一行显示两个字符，movl后第一串字符表示具体要输出的数据的形式与内容，逗号后表示地址。然后align表示以4字节对齐。然后再multiboot\_header中写入multiboot协议的内容，最后在写入一个死循环。

**代码布局说明：**

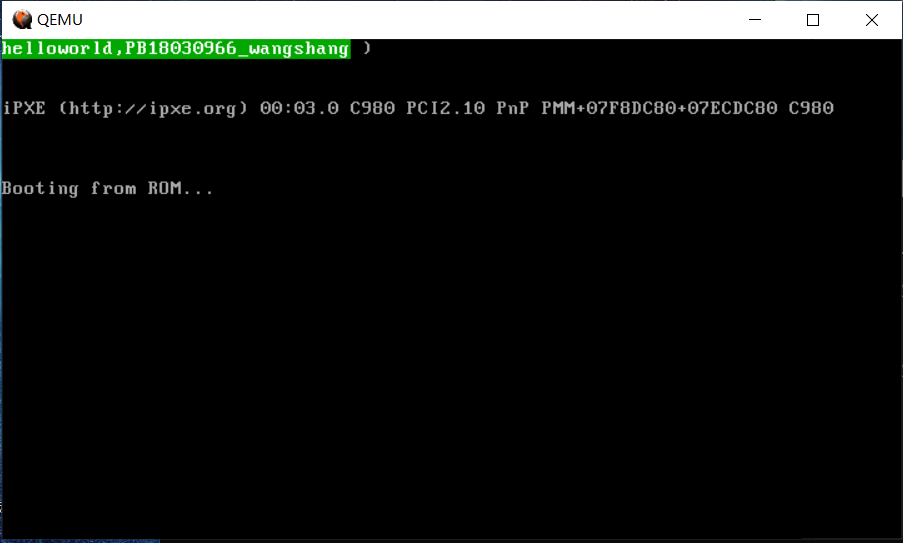
**编译过程说明：**

在Ubuntu中先搜索到workspace目录，然后通过指令make完成编译，可以看到在workspace目录下获得了bin文件与o文件。





**运行和运行结果说明：**

在Ubuntu中通过QEMU启动已经编译生成的bin文件，通过了multiboot协议后才能运行，于是得到Linux的图形化界面运行结果，显示需要的输出。其中输出结果采用了绿底白字，正好对应了在汇编文件中的对输出数据的格式的说明。

**遇到的问题和解决方案：**

1. 不知道如何运用Linux命令？通过查阅网上资料解决。
2. 不会编写汇编文件？仔细浏览并借鉴了multiboot协议相关内容说明后成功编译。
3. 因为无法运行SDL而无法运行内核？翻阅一定资料、求助同学后发现在课程讨论上发现了相关内容并解决。