

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 VGA 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。

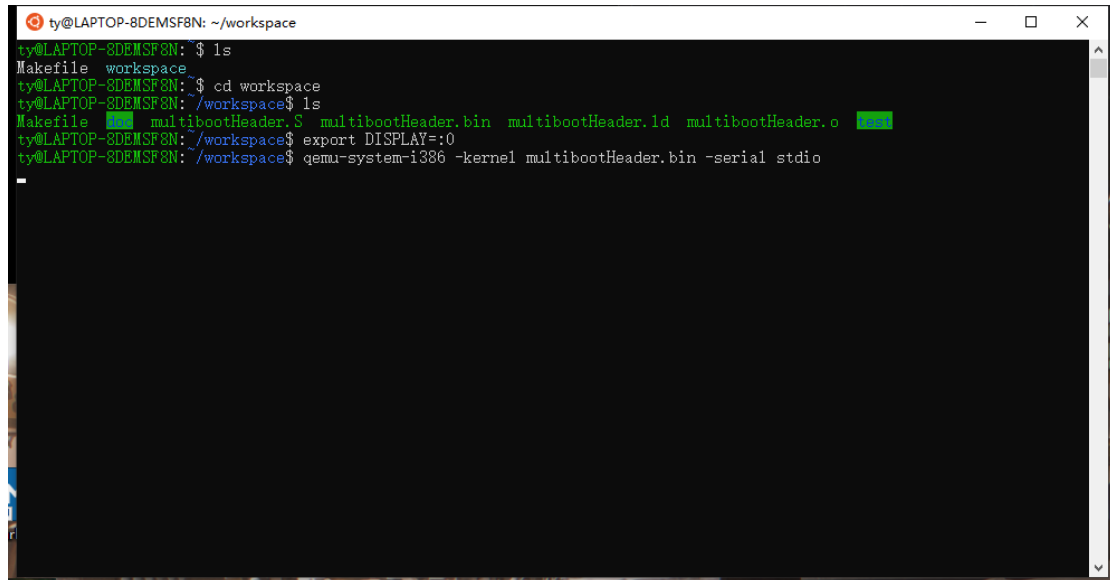
S 文件说，先声明 start、_start 的全局性，然后在 start 中写入需要在 VGA 上显示的内容，其中 movl 就是在显存上写入需要显示的数据，每一行显示两个字符，movl 后第一串字符表示具体要输出的数据的形式与内容，逗号后表示地址。然后 align 表示以 4 字节对齐。然后再 multiboot_header 中写入 multiboot 协议的内容，最后在写入一个死循环。

代码布局说明：

在 ld 文件中可见，其中 Section 表示物理内存地址从 1M 处开始，即跳过了 1M 的大小，而代码布局以 8 字节对齐。

编译过程说明：

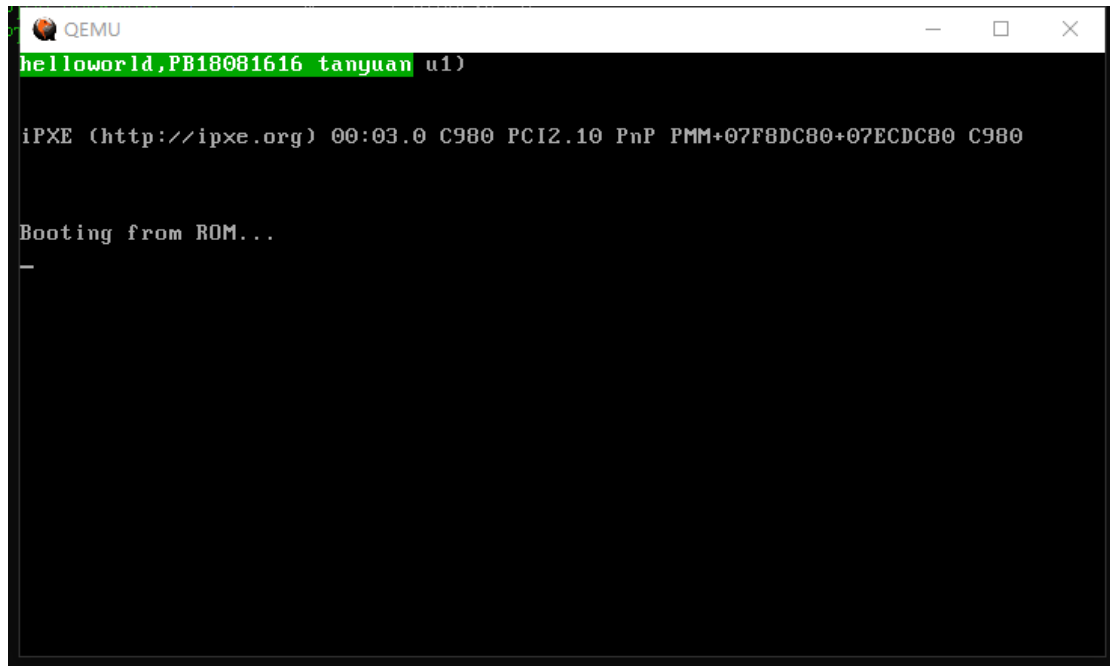
在 Ubuntu 中先搜索到 workspace 目录，然后通过指令 make 完成编译，可以看到在 workspace 目录下获得了 bin 文件与 o 文件。



```
ty@LAPTOP-8DEMSF8N: ~/workspace
ty@LAPTOP-8DEMSF8N:~$ ls
Makefile workspace
ty@LAPTOP-8DEMSF8N:~$ cd workspace
ty@LAPTOP-8DEMSF8N:~/workspace$ ls
Makefile  multibootHeader.S  multibootHeader.bin  multibootHeader.ld  multibootHeader.o
ty@LAPTOP-8DEMSF8N:~/workspace$ export DISPLAY=:0
ty@LAPTOP-8DEMSF8N:~/workspace$ qemu-system-i386 -kernel multibootHeader.bin -serial stdio
```

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说明。



遇到的问题和解决方案：

1. 不会编写汇编文件？仔细浏览了 ppt 后在网上相关资料的帮助下勉强能编译成功了，
2. 因为无法运行 SDL 而无法运行内核？在提交说明中找到了解决办法。