**操作系统原理与设计 实验报告**

姓名：孙昊哲 学号：PB20000277 实验日期：2022-3-15

**一、实验题目：**

实验一：启动

**二、原理说明：**

# 启动协议：

# Multiboot协议：

这个规范主要瞄准PC，因为它们最普遍，并且具有最多种多样的操作系统及启动载入器。不过，某些其它架构可能需要一个启动规范，但还没有现成的，这个规范的一个修改，剥除特定于x86的细节，也可以适用于它们。

这个规范针对自由32位操作系统，这些系统可以相当容易地被修改来支持这个规范，不需要许多官僚的繁文缛节。这个规范主要为之设计的自由操作系统是，Linux，FreeBSD及NetBSD的内核。

Multiboot-compliant OS kernels:应该使用ELF格式。并且头文件至少需要有三个参数，分别为magic（0x1BADB002），flags（可以是0），checksum（magic + checksum + flags =0）。

# QEMU：

# QEMU（quick emulator）是一款由法布里斯·贝拉（Fabrice Bellard）等人编写的免费的可执行硬件虚拟化的（hardware virtualization）开源托管虚拟机（VMM）。其与Bochs，PearPC类似，但拥有高速（配合KVM），跨平台的特性。

# QEMU是一个托管的虚拟机镜像，它通过动态的二进制转换，模拟CPU，并且提供一组设备模型，使它能够运行多种未修改的客户机OS，可以通过与KVM一起使用进而接近本地速度运行虚拟机（接近真实电脑的速度）。

# QEMU还可以为user-level的进程执行CPU仿真，进而允许了为一种架构编译的程序在另外一种架构上面运行（借由VMM的形式）。

# VGA：

# 视频图形阵列（英语：Video Graphics Array，缩写：VGA）是IBM的一个使用模拟信号的电脑显示标准，在1987年随IBM PS/2系列计算机推出。VGA是大多数PC制造商所遵循的最后一个IBM图形标准，几乎1990年后的所有PC图形硬件都最低支持VGA。

# 本实验中字符界面规格：25行80列；

# VGA显存的起始地址：0xB8000；

# 直接写VGA显存可以输出信息。

# 串口：

# 在qemu中，即使MyOS不对uart进行初始化，也可以正确输出：

# 端口地址：0x3F8

# 输出字符的汇编代码为：

# movb $0x46, %al /\* 输出字母F \*/

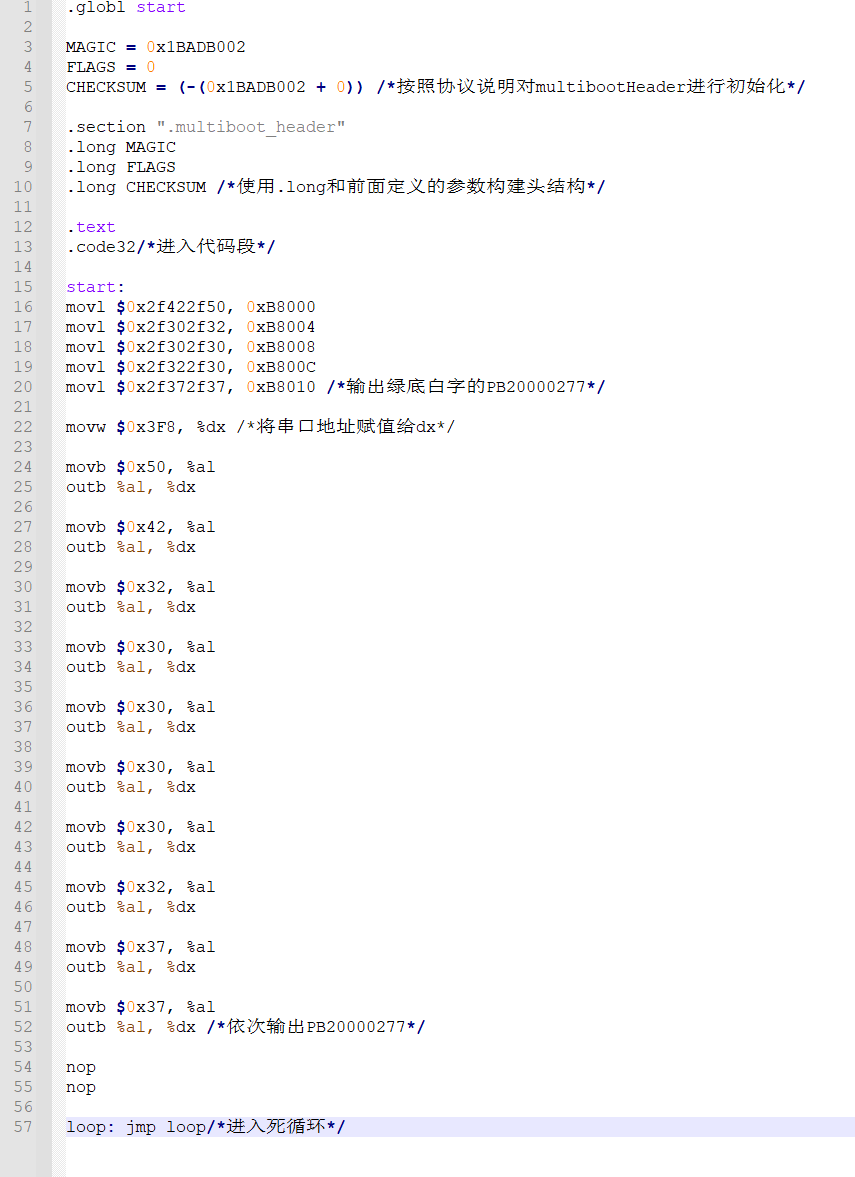
# movw $0x3F8, %dx

# outb %al, %dx /\* put char \*/

# nop

# nop

# 三、源代码说明：



本代码 首先按照协议说明对multibootHeader进行初始化；再使用.long和前面定义的参数构建头结构；进入代码段后，将需要输出的颜色以及ASCII码赋值给VGA地址，VGA正确输出绿底白字的PB20000277;将需要输出的ASCII码赋值给所需要的变量，然后使用outb命令端口输出PB20000277，最后程序进入死循环。

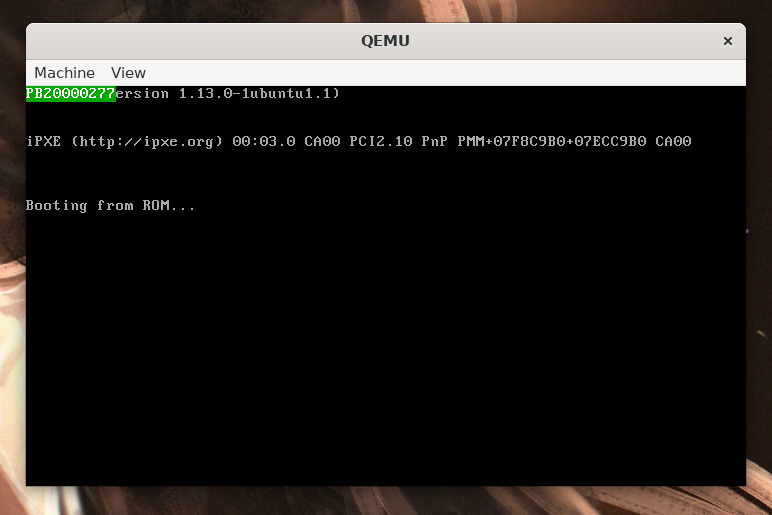
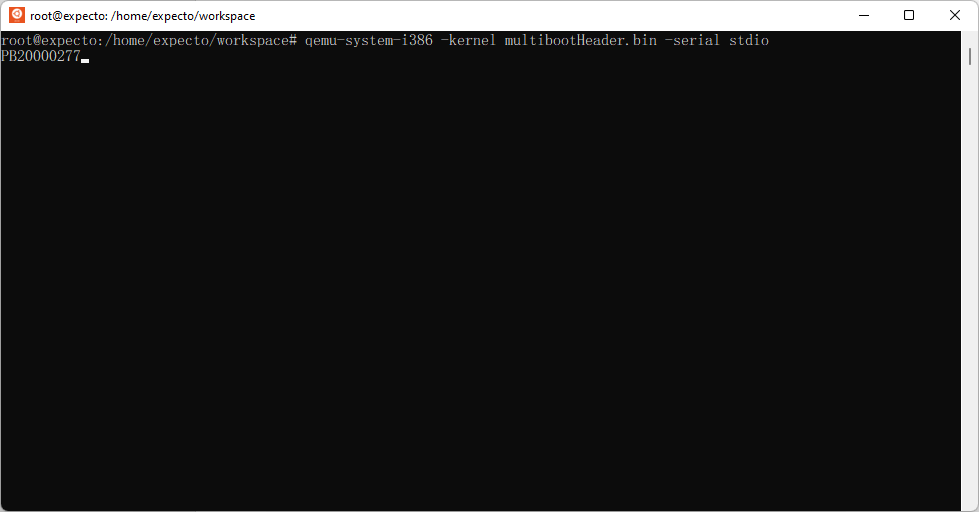
**四、代码布局说明：**

本篇代码，VGA部分地址从0xB8000开始，并且每次向VGA地址传输需要输出的字符颜色及ASCII码。共32字节，所以每次VGA地址加4。

**五、编译过程说明：**

本次实验在Windows11中的Linux子系统（Ubuntu 20.04.4LTS）中编译完成的。其中Makefile文件和MultibootHeader.S文件由老师提供。本次编译gcc编译器更新到最新，使用gcc编译完成，并由ld命令链接后，最终生成multibootHeader.bin文件。

**六、运行和运行结果说明：**



使用qemu-system-i386 -kernel multibootHeader.bin -serial stdio命令在Ubuntu系统中正常运行。

并且结果符合预期，其中在VGA输出端口有一些QEMU启动时自动生成的字符，符合正常预期，老师也并没有要求我们执行清屏操作。

**七、遇到的问题和解决方案说明：**

（1）本次实验在编译的过程中发现没有安装gcc模块，使用指令apt install gcc后安装成功。

（2）在Vlab实验环境中提示没有连接到dbus，但是可以正常输出，并且在询问助教及同学，及网上查阅资料后未果，后转到WSL平台，一切正常。