中国科学技术大学计算机学院

《操作系统原理与设计》

实验题目：multiboot启动

学生姓名：叶子昂

学生学号：PB20020586

完成时间：2022年3月8日

0.1 实验题目

multiboot启动

0.2 实验目的

编写multiboot header实现以下功能

- 在屏幕实现输出特定内容
- 串口输出特定内容

0.3 实验环境

windows11，wsl2，qemu

0.4 实验原理

- multiboot启动：

> multiboot启动须遵守multiboot协议编写正确的multibootheader其中magic，flags，checksum是必须的它们均是u32类型的数。

- VGA输出：

> 按照VGA输出规则(ppt所给)直接在正确的位置写入显存即可

- 串口输出：

> 向%dx寄存器写入串口端口，向%al写入输出内容，使用outb指令输出即可

0.5 代码结构与编译运行

0.5.1 代码分析

1. multibootheader编写

   根据multiboot协议magic，flags，checksum是必须的。根据协议介绍magic值为0x1BADB002，本实验暂不需要设置flag的各位设为0X00000000，checksum与magic相加应为0，故其为-0X1BADB002。设置它们的值并声明，代码如下。

```
/*定义协议要求的参数*/
magic_ITEM_NAME = 0X1BADB002;#magic
flag_ITEM_NAME = 0X00000000;#flag
check_ITEM_NAME = -0X1BADB002;#check

.section ".multiboot_header"#对应link文件的文件名
.align 4
.long magic_ITEM_NAME
.long flag_ITEM_NAME
.long check_ITEM_NAME
/*与前面参数构建头结构*/
```

2. 代码段VGA输出

按照VGA输出规则一次可以输出一个字符同时占两个字节共16位，movl指令可以向显存写入32位4字节内容，故一条movl指令包含两个VGA输出占4字节显存，两条movl指令目的显存相差4字节，起始显存为0x8000 。设置输出格式为白底绿字f2，代码如下：

```
/*the output of VGA*/
movl $0xf242f250, 0xB8000 #学号PB20020586
movl $0xf230f232, 0xB8004
movl $0xf232f230, 0xB8008
movl $0xf235f230, 0xB800C
movl $0xf236f238, 0xB8010
movw $0xf200, 0xB8014       #下划线
movl $0xf265f268, 0xB8016 #hello,myos!
movl $0xf26cf26c, 0xB801A
movl $0xf22cf26f, 0xB801E
movl $0xf279f26d, 0xB8022
movl $0xf253f24f, 0xB8026
movl $0xf200f221, 0xB802A
movl $0xf200f200, 0xB802E #覆盖原有内容(ubuntu...)
movl $0xf200f200, 0xB8032
movl $0xf200f200, 0xB8036
movl $0xf200f200, 0xB803A
movl $0xf200f200, 0xB803E
movl $0xf200f200, 0xB8042
movl $0xf200f200, 0xB8046
movl $0xf200f200, 0xB804A
movl $0xf200f200, 0xB804E
movl $0xf200f200, 0xB8052
movl $0xf200f200, 0xB8056
```

3. 代码段串口输出

向%dx寄存器写入串口端口0x3F8，向%al写入输出字符的ASCII码，使用outb指令输出即可

```
/*the output of uart*/
movb $0x68, %al                 #hello,yeziang
movw $0x3F8, %dx
outb %al, %dx
movb $0x65, %al
outb %al, %dx
movb $0x6c, %al
outb %al, %dx
movb $0x6c, %al
outb %al, %dx
movb $0x6f, %al
outb %al, %dx
movb $0x2c, %al
outb %al, %dx
movb $0x79, %al
```

```
    outb %al, %dx
    movb $0x65, %al
    outb %al, %dx
    movb $0x7a, %al
    outb %al, %dx
    movb $0x69, %al
    outb %al, %dx
    movb $0x61, %al
    outb %al, %dx
    movb $0x6e, %al
    outb %al, %dx
    movb $0x67, %al
    outb %al, %dx
```

4. 其他

   .globl start声明使其他外部文件能使用start标识

   结尾的hlt停机指令阻止重复输出

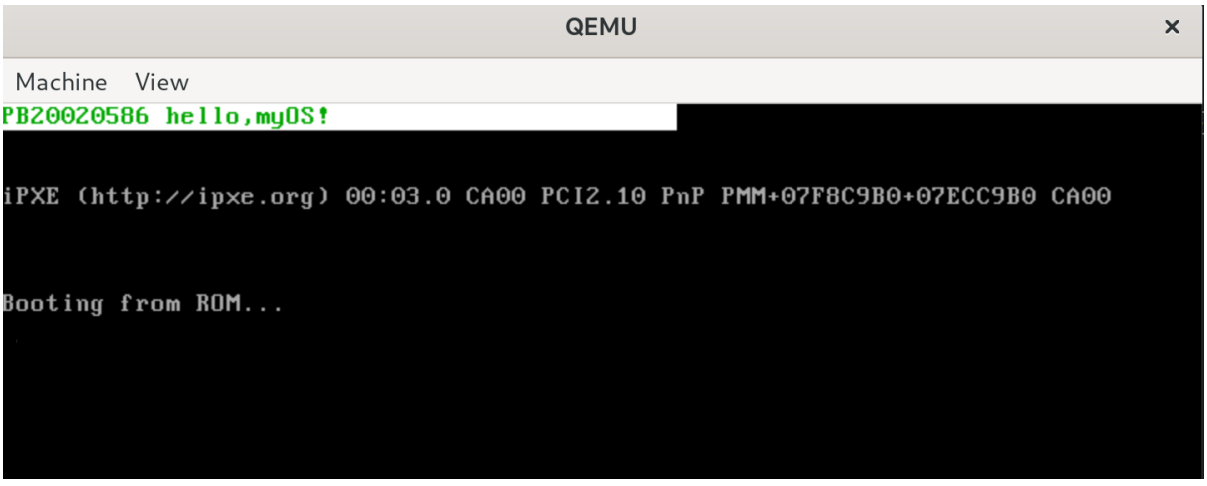
0.5.2 编译运行

- 编译：根据makefile定义的规则按照multibootheader.ld提供的连接方法编译，在终端输入make指令即可。
- 运行：在qemu模拟的环境下运行，工作目录下输入以下命令即可

  > qemu-system-i386 -kernel multibootHeader.bin -serial stdio

- 运行结果截图

  VGA输出：



  串口输出：

0.6 遇到的问题及解决

1. sudo apt install qemu后仍无法使用qemu-system-i386命令.

   > 解决:使用sudo apt install qemu-system-x86

2. 串口输出重复，VGA输出闪烁

   > 解决：在代码段末尾加入hlt停机指令

0.7 总结与思考

- 在本次实验的环境准备过程中，我进一步熟悉了解linux环境和常用命令
- 通过本次实验我简单了解了AT&T汇编语言，能根据ppt所给示例利用VGA和串口进行输出
- 通过阅读multibootheader协议简要了解了multiboot的启动原理，能够正确编写multibootheader在qemu环境下启动。