中国科学技术大学计算机学院

《操作系统原理与设计实验》Lab_01 报告

实验题目：Multiboot 启动

学生姓名：赵卓

学生学号：PB21111686

完成日期：2024 年 3 月 14 日

【实验题目】

Multiboot 启动

【实验目的】

实现一个最初的操作系统，包括 Multiboot 启动头和最简单的操作系统内核。

【实验内容】

补全 multibootHeader.S 汇编文件，启动操作系统内核，在屏幕

和串口上输出指定内容。

【实验原理】

Multiboot：Multiboot 启动协议是一套多种操作系统共存时的启动引导协议。现今，许多操作系统都拥有自己的 boot loader（即引导程序，负责载入最终的操作系统映像的一个或一组程序），若要使多种操作系统在机器上共存,则需要借助于特定的引导机制,显然,这很可能对使用多操作系统的便利性与兼容性带来影响。为解决这一问题，Multiboot 启动协议应运而生。简单来讲，这份协议明确了 boot loader 与操作系统间的接口,所有符合协议规范的 boot loader 可以读取所有所有符合协议规范的操作系统。

Qemu：qemu 是一个虚拟化模拟器，使用它可以对一些常用的处理器进行模拟。本实验中 QEMU 作用为 OS 运行提供系统模拟硬件，比如虚拟 VGA 和串口。

VGA：VGA 是一种标准显示接口，可以在其显存中写入内容输出。

串口：通过串口可以一位位输出数据。

本实验通过编写 multibootHeader 汇编文件 make 生成最小的 OS 内核，并通过 qemu 模拟运行，输出指定内容。

【代码说明】

Multiboot 参数定义：按协议要求定义。

```
.globl start # 一般都用start
MAGIC_ITEM_NAME = 0x1BADB002
FLAGS_ITEM_NAME = 0x00000000
CHECKSUM_ITEM_NAME = -(MAGIC_ITEM_NAME + FLAGS_ITEM_NAME)
```

Muiltiboot 协议头：按协议要求定义。

```
/*此处开始，按协议标准来定义必须的multiboot header*/
.section ".multiboot_header"
.long MAGIC_ITEM_NAME
.long FLAGS_ITEM_NAME
.long CHECKSUM_ITEM_NAME
```

VGA 输出：将内容字符的 ASCII 码按照顺序依次存入 VGA 显存。

```
.text # 进入代码段
.code32 # 32位代码
 # 这个跟第一行的声明要一致
start:
    vga:
        movl $0x2f652f48, 0xB8000    #He
        movl $0x2f6c2f6c, 0xB8004    #ll
        movl $0x2f202f6f, 0xB8008    #o' '
        movl $0x2f6f2f77, 0xB800C    #wo
        movl $0x2f6c2f72, 0xB8010    #rl
        movl $0x2f212f64, 0xB8014    #d!
        movl $0x2f502f20, 0xB8018    #' 'P
        movl $0x2f322f42, 0xB801C    #B2
        movl $0x2f312f31, 0xB8020    #11
        movl $0x2f312f31, 0xB8024    #11
        movl $0x2f382f36, 0xB8028    #68
        movl $0x2f5f2f36, 0xB802C    #6_
        movl $0x2f682f7A, 0xB8030    #zh
        movl $0x2f6F2f61, 0xB8034    #ao
        movl $0x2f682f7A, 0xB8038    #zh
        movl $0x2f6F2f75, 0xB803C    #uo
```

串口输出：和 VGA 不同，串口输出通过寄存器%al 将内容传递至串口端口地址输出。

```
/*根据需要初始化串口*/
/*根据需要串口输出你的字符序列，详见前面串口编程简介*/
    uart:
        movw $0x3F8, %dx
        movb $0x48, %al  #H
        outb %al, %dx
        movb $0x65, %al  #e
        outb %al, %dx
        movb $0x6c, %al  #l
        outb %al, %dx
        movb $0x6c, %al  #l
        outb %al, %dx
        movb $0x6f, %al  #o
        outb %al, %dx
        movb $0x20, %al  #' '
        outb %al, %dx
        movb $0x77, %al  #w
        outb %al, %dx
        movb $0x6f, %al  #o
        outb %al, %dx
        movb $0x72, %al  #r
        outb %al, %dx
        movb $0x6c, %al  #l
        outb %al, %dx
        movb $0x64, %al  #d
        outb %al, %dx
        movb $0x21, %al  #!
        outb %al, %dx
        movb $0x20, %al  #' '
        outb %al, %dx
        movb $0x50, %al  #P
        outb %al, %dx
        movb $0x42, %al  #B
        outb %al, %dx
        movb $0x32, %al  #2
        outb %al, %dx
        movb $0x31, %al  #1
        outb %al, %dx
        movb $0x31, %al  #1
        outb %al, %dx
        movb $0x31, %al  #1
        outb %al, %dx
        movb $0x31, %al  #1
        outb %al, %dx
        movb $0x36, %al  #6
        outb %al, %dx
        movb $0x38, %al  #8
        outb %al, %dx
        movb $0x36, %al  #6
        outb %al, %dx
        movb $0x5f, %al  #_
        outb %al, %dx
        movb $0x7A, %al  #z
        outb %al, %dx
        movb $0x68, %al  #h
        outb %al, %dx
        movb $0x61, %al  #a
        outb %al, %dx
        movb $0x6F, %al  #o
        outb %al, %dx
        movb $0x7A, %al  #z
        outb %al, %dx
        movb $0x68, %al  #h
        outb %al, %dx
        movb $0x75, %al  #u
        outb %al, %dx
        movb $0x6F, %al  #o
        outb %al, %dx
/*实验结束，让计算机停机，方法：使用hlt指令，或者死循环*/
    hlt
```

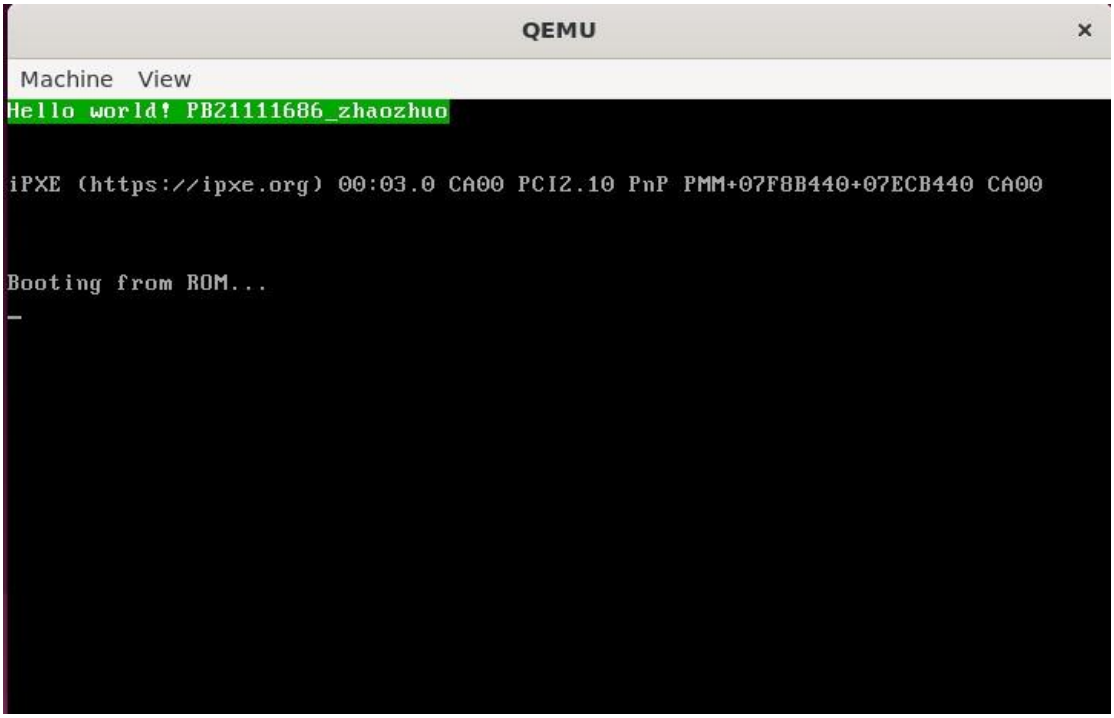
Makefile：一种自动化构建程序的文件，类似于 bash 脚本，make 后自动执行指令，在 ld 描述文件下，生成 bin 内核。

```
ASM_FLAGS = -m32 --pipe -Wall -fasm -g -O1 -fno-stack-protector

multibootHeader.bin: multibootHeader.S
    gcc -c ${ASM_FLAGS} multibootHeader.S -o multibootHeader.o
    ld -n -T multibootHeader.ld multibootHeader.o -o multibootHeader.bin

clean:
    rm -rf ./multibootHeader.bin ./multibootHeader.o
```

【实验结果】

Make 后通过 qemu 模拟执行得到如下串口和 VGA 输出结果，符合要求：

```
zz@zz:~/src$ qemu-system-i386 -kernel multibootHeader.bin -serial stdio
Hello world! PB21111686_zhaozhuo
```



【总结与思考】

本次实验了解了 Multiboot 协议，汇编，makefile 等多个知识，编写了一个最简单的 OS 内核。遇到的错误是 VGA 地址计算错误,将"3C"写成"4C"，改正后即输出正确。