

Lab1 实验文档

实验目标

实现multiboot header，启动OS内核，并在VGA和串口输出一些内容。

框架图

```
|-- doc
|   |-- report.pdf          <- 要提交的实验报告
|-- src
|   |-- Makefile            <- 已经给出
|   |-- multibootHeader.S   <- 你要填写
|   |-- multibootHeader.bin <- 编译后生成
|   |-- multibootHeader.ld  <- 已经给出
```

任务

1. 配置相关环境

WSL 是什么？

WSL 是windows subsystem for linux的简写，指的是windows的一个子系统，这个子系统的作用是在windows下运行linux操作系统。更多请查看[微软WSL文档](#)

QEMU 是什么？

QEMU is a generic and open source machine emulator and virtualizer.

QEMU can be used in several different ways. The most common is for System Emulation where it provides a virtual model of an entire machine (CPU, memory and emulated devices) to run a guest OS

上述摘自[QEMU官方文档](#)，本实验中QEMU作用是为OS运行提供系统模拟硬件，比如虚拟VGA和串口。

QEMU 和 WSL 的关系

WSL在本次实验中为我们提供Linux 开发环境。QEMU为我们写的操作系统提供模拟硬件

multiboot 是什么？

在计算机启动过程中，需要由bootloader来加载操作系统。

multiboot的产生背景是不同的操作系统倾向于选择不同的 bootloader(引导加载程序)，在多系统的情况下，情况往往会变得很复杂，很可能会导致系统运行失败。

multiboot的出现就是为了解决这一问题。**multiboot本质是一个协议**，它能够指定bootloader和操作系统之间的接口，以便任何符合的引导加载程序都应该能够加载任何符合的操作系统。

如果能够编写一个支持multiboot协议的操作系统，无论它有多小，multiboot都会为其找到合适的bootloader 加载启动操作系统内核。

更多请查看[Multiboot Specification version 0.6.96](#)

配置方案

- 方案一：WSL + QEMU
 - WSL + VcXsrv（可视化工具）：[教程](#)
 - 为方便操作，推荐按PPT中方法建立**共享文件夹**
 - 换源：WSL默认源下载QEMU可能会出错，所以需要为WSL进行换源：[教程](#)
 - QEMU：Ubuntu命令行中输入指令 `sudo apt-get install qemu`
 - 下载其他软件：看Makefile文件想想还需要下什么
- 方案二：VMware + QEMU
 - VMware：查看文档《windows安装VMware教程》

其他方案可以自己尝试配置

2. 填写三个文件

a. Makefile 和 .ld 文件

已经给出，也可以根据需求修改。但要求理解两个文件中的内容和作用。

b. 编写 .s 文件

PPT中给出的部分汇编代码

```
.globl start #一般都用start
/*此处，按照multiboot协议要求定义必要的参数*/
/*格式: XXX_ITEM_NAME=value*/
/*value可以是十六进制的（0x开头）、十进制的等等*/

/*此处开始，按协议标准来定义必须的multiboot header*/
.section ".xxx" #先起一个section的名字
/*使用.long和前面定义的参数构建头结构，每次32位，格式为: .long XXX_ITEM_NAME*/

.text #进入代码段
.code32 #32位代码
    start: #这个跟第一行的声明要一致
    /*下面屏幕输出OK或其他字符序列*/
    /*使用指令movl $0x12345678, 0xB8000*/ #0x1234和0x5678各自输出1个字符
    /*根据需要输出多个字符，也可以使用其他mov指令*/
    /*可以根据需要使用nop指令隔开不同功能片段，可以适当使用空行*/
    /*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*/
    /*实验结束，让计算机停机，方法：使用hlt指令，或者死循环*/
```

提示：

- 定义multiboot 必要参数部分是PPT中magic、flags、checksum 参数。如何设置见PPT或者在[multiboot协议](#) 3.1.1 和 3.1.2中查看
- .section 按照注释格式编写即可

- .text 部分进行VGA和串口（选做）输出。
 - VGA仿照注释中指令将输出的字符输出即可。可能会使用 `mov` 指令的其他格式，如 `movw`
 - 串口（UART）参照PPT中“串口编程简介”中伪代码即可

本实验使用x86汇编风格，指令详细规定请查阅相关文档

3. 编译运行

使用 `make` 指令生成 `.bin` 文件

使用 `qemu-system-i386 -kernel multibootHeader.bin -serial stdio` 来进行运行

指令详解：

`qemu-system-i386` 指定平台为i386

`-kernel` 指定内核文件

`-serial stdio` 指定串行终端为标准输入输出

拓展：

终端选项（`-serial --parallel --monitor`）

终端包括三类：串行终端，并行终端，`qemu`命令监控终端。

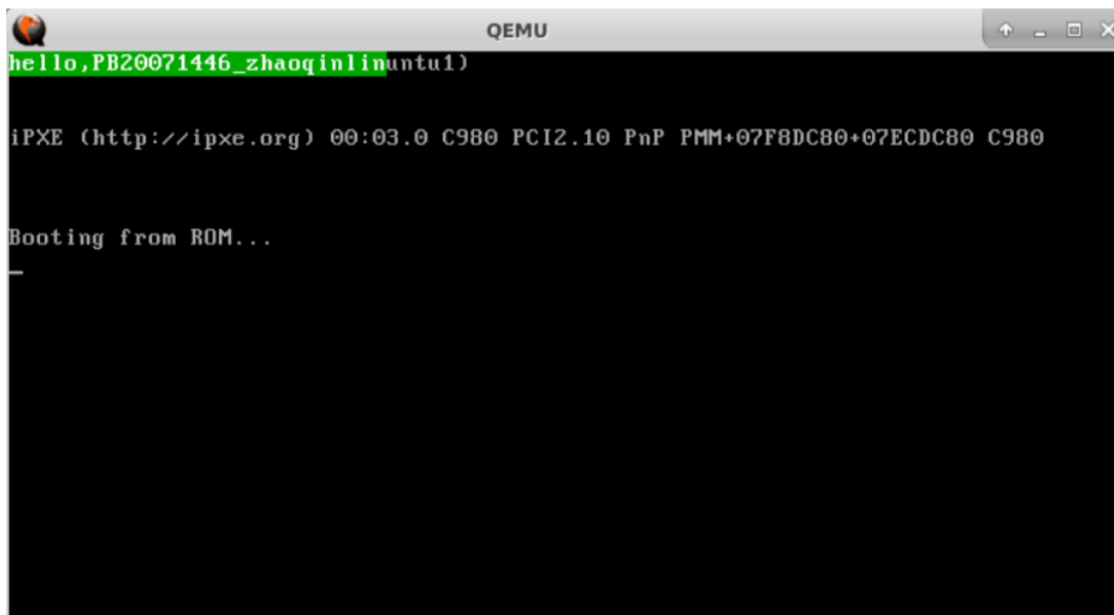
如果运行QEMU后会发现无法拖动鼠标到QEMU显示屏外的区域，请使用 `Ctrl + Alt` 解决此问题

最终效果图如下

1.串口输出zql

```
matrix3@LAPTOP-SRSJ55M6:/mnt/c/Users/31363/Desktop/workspace$ qemu-system-i386  
kernel multibootHeader.bin -serial stdio  
zql
```

2.VGA输出



提交要求

截止时间：2022.3.28晚23.59分！逾期不接受补交。

提交内容

- 提交平台：BB平台对应板块
- 提交方式：按照上面的框架图组织文件，将其**打包压缩**（压缩格式不限），命名为学号_姓名_lab1。并且将report**单独**再提交一次。
- 也就是说你最终要上传至BB系统的为 学号_姓名_lab1.zip和report.pdf这两个文件

实验要求

- 完成汇编文件multibootHeader.S
- 运行make指令能够成功生成multibootHeader.bin
- 能够使用qemu启动，并且成功实现vga输出，串口选做

你所显示的内容应具有与你相符的特异性，类似"PB20211111_huaqiang"；不具有特异性或者与他人相同的显示将被视为抄袭！

实验报告要求

- 需要简单对实验原理进行描述（无需对ppt内容进行复制粘贴）
- 需要对代码运行方法、编译过程进行说明
- 需要对你的源代码和地址空间分配等做出一定解释，并且在**报告中附上你的显示效果截图**
- 可以对自己在实验中所遇到的问题和解决方法进行描述（不做硬性要求）

其他要求

代码风格在cs学习过程中是至关重要的一部分，有合理缩进、标识符安排，以及适量有用的注释等等，都是良好的习惯。我们会给予一定分数奖励

评分标准

总分100分

| 评价项目 | 最高分值 |
|--------|------|
| 能编译运行 | 50 |
| 实验报告 | 40 |
| 实现串口输出 | 5 |
| 代码风格 | 5 |

反馈

本实验文档由赵钦林助教编写，VMware教程由高旭民助教编写。如有问题或建议可向相关助教进行反馈。非常感谢！

