实验名称：lab1_startup

1. 原理说明：

Qemu 软件提供了一个模拟的 i386 架构硬件环境，我们自己编写一个源代码并且自己编译链接生成最终的内存映像文件，使此文件能够按照源代码里规定的方式在 qemu 提供的内存上跑起来

2. 源代码说明：

1.Makefile:

```
ASM_FLAGS= -m32 --pipe -Wall -fasm -g -O1 -fno-stack-protector
multibootHeader.bin: multibootHeader.S
    gcc -c ${ASM_FLAGS} multibootHeader.S -o multibootHeader.o
/*将 ASM_FLAGS 作为参数，编译 multibootHeader.S 文件，并生成 multibootHeader.o 文件*/
    ld -n -T multibootHeader.ld multibootHeader.o -o multibootHeader.bin
/*用 ld 文件和.o 文件作为参数编译链接生成.bin 文件*/
    qemu-system-i386 -kernel multibootHeader.bin -serial stdio -M pc-i440fx-3.1
/*用 qemu 的 i386 内核运行.bin 文件*/
clean:
 rm -rf ./multibootHeader.bin ./multibootHeader.o
/*删除旧的.bin/.o 文件以便生成新的.bin/.o 文件
```

2.multiheader.ld

```
OUTPUT_FORMAT("elf32-i386", "elf32-i386", "elf32-i386")//输出格式
OUTPUT_ARCH(i386)//输出架构
ENTRY(start)//入口标记，与.s 文件对应
SECTIONS {
. = 1M;//分配的空间大小
.text : {
*(.multiboot_header)//执行代码段标记
. = ALIGN(32);//对其格式
*(.text)
}
}
```

3. multibootHeader.S:

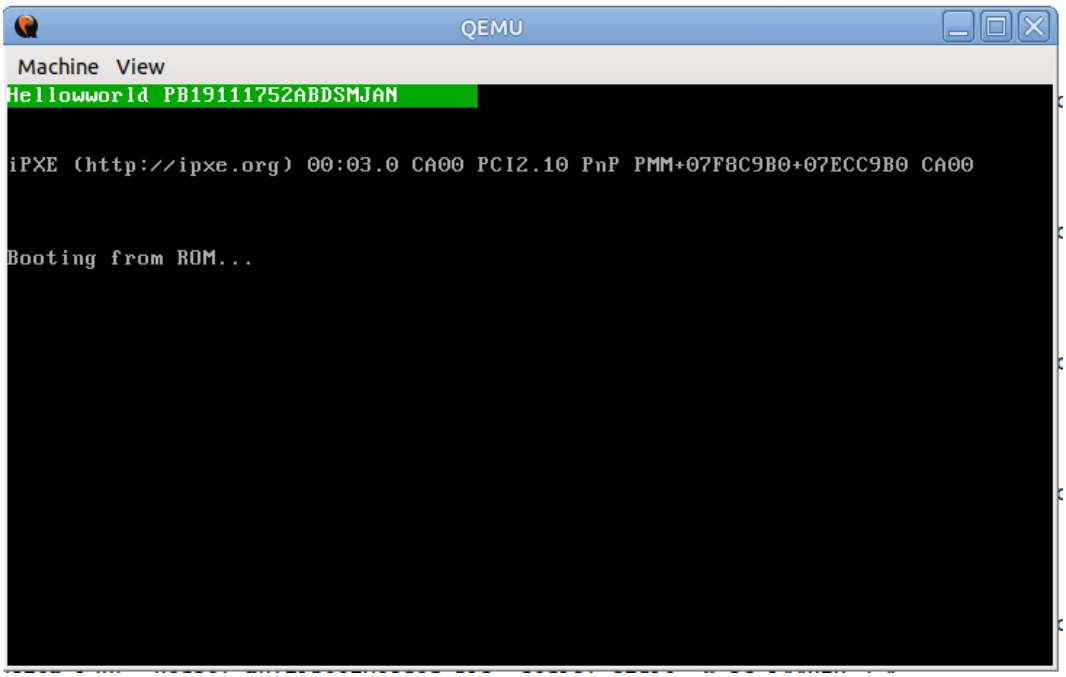
```
.set magic,0x1BADB002                    //设定 magic 变量，一下三段雷同
.set flags,0
```

```
.set checksum,-(magic+flags)

.section .multiboot_header          //执行代码段片段
.ALIGN 32                           //对其格式
.long magic                         //和前面定义的参数构成头结构
.long flags
.long checksum

.section .text                      //代码语句执行段
.globl start                        //代码执行入口，和.ld 文件对应
start:

    movl $0x2f652f48, 0xB8000       //VGA 输出相应的字符到指定的内存
地址上
    movl $0x2f6c2f6c, 0xB8004
    movl $0x2f772f6f, 0xB8008
    movl $0x2f6f2f77, 0xB800C
    movl $0x2f6c2f72, 0xB8010
    movl $0x2f202f64, 0xB8014
    movl $0x2f422f50, 0xB8018
    movl $0x2f392f31, 0xB801C
    movl $0x2f312f31, 0xB8020
    movl $0x2f372f31, 0xB8024
    movl $0x2f322f35, 0xB8028
    movl $0x2f422f41, 0xB802C
    movl $0x2f532f44, 0xB8030
    movl $0x2f4a2f4d, 0xB8034
    movl $0x2f4e2f41, 0xB8038
    movl $0x2f202f20, 0xB803C
    movl $0x2f202f20, 0xB8040
    movl $0x2f202f20, 0xB8044

  hlt
```

4. 编译过程说明：

源代码配上 makefile 里头指定的参数进行编译生成.o 文件,.o 文件和.ld 文件作为参数编译链接生成.bin 文件，.bin 文件就是我们想要的最终文件。

5. 运行结果说明：



6. 遇到的问题和解决方案：

在运行 qemu-system-i386 -kernel multibootHeader.bin -serial stdio 命令时 遇到了“.....pvm kernel.....”问题 此时助教要求我更换 Ubuntu 版本到 LTS18.0 的版本 ，但是在那个版本里遇到没有图形化库的问题，然后又回到 vlab.ustc.edu.cn 里的虚拟机继续做，最终从网上查阅到在上面的命令后面加个参数之前的问题就可以解决即” qemu-system-i386 -kernel multibootHeader.bin -serial stdio -M pc-i440fx-3.1” 然后就可以顺利的进入到了 qemu 的系统模拟环境里，最后由于遇到，输出混乱的问题，发现是因为没有注意到 16 进制加法问题，最终此问题也得到解决。总体完成过程很并不容易。