任宇轩PB20030736lab1

实验原理

multiboot

基于multi boot协议**Multiboot Specification version 0.6.96**编写 简单系统头 bootloader *ELF format*

> 编写一个*bootloader* 为其提供存储空间和其他资源
>
> *Multiboot-compliant OS images* 必须包含一个固定格式的 *multi bootheader*

再用qemu 来运行multiboot 进行编写好的输出进程

VGA

用来实现系统与显卡输出交流的接口 为阵列格式 ， 通过对特定的坐标输出内容 得到特定的image

串口

串行接口输出 Serial Interface 指数据一位一位的顺序传送

编写文件头时因为输出的接口 固定 所以串口输出只需重复同一地址访存以达到串口输出

源代码说明

Makefile

```
ASM_FLAGS= -m32 --pipe -Wall -fasm -g -O1 -fno-stack-protector #定义宏
multibootHeader.bin: multibootHeader.S
gcc -c ${ASM_FLAGS} multibootHeader.S -o multibootHeader.o #对汇编文件编译生成可重定位文件
ld -n -T multibootHeader.ld multibootHeader.o -o multibootHeader.bin #对文件进行重定位 链接生成 bin
clean:
rm -rf ./multibootHeader.bin ./multibootHeader.o #删除文件
```

multibootHeader.ld

```
OUTPUT_FORMAT("elf32-i386", "elf32-i386", "elf32-i386") #定义文件格式
OUTPUT_ARCH(i386) #架构
ENTRY(start) #进入入口段：'start'
SECTIONS {
. = 1M;
.text : {
*(.multiboot_header)
. = ALIGN(8);
*(.text)
}
}
#地址空间大小为1M 由两个段组成 .multiboot_header & .text
```

multibootHeader.s

```
.globl start

MULTIBOOT_HEADER_MAGIC=0x1BADB002
MULTIBOOT_HEADER_FLAGS=0
MULTIBOOT_HEADER_CHECKSUM=0xE4524FFE

.section multiboot_header
.align 4
.long MULTIBOOT_HEADER_MAGIC
.long MULTIBOOT_HEADER_FLAGS
.long MULTIBOOT_HEADER_CHECKSUM
.text #进入代码段
.code32 #32位代码

start:# vga输出
movl $0x2f652f48, 0xB8000
movl $0x2f6c2f6c, 0xB8004
movl $0x2f6f, 0xB8008
movl $0xce6fce77, 0xB8012
movl $0xce6cce72, 0xB8016
movl $0xce64, 0xB801A
#VGA: hello world

movl $0x3c423c50, 0xB80A0
movl $0xfc30fc32, 0xB80A4
movl $0xfc33fc30, 0xB80A8
movl $0xfc37fc30, 0xB80AC
movl $0xfc36fc33, 0xB80B0

#VGA:学号输出

# 串口输出
```

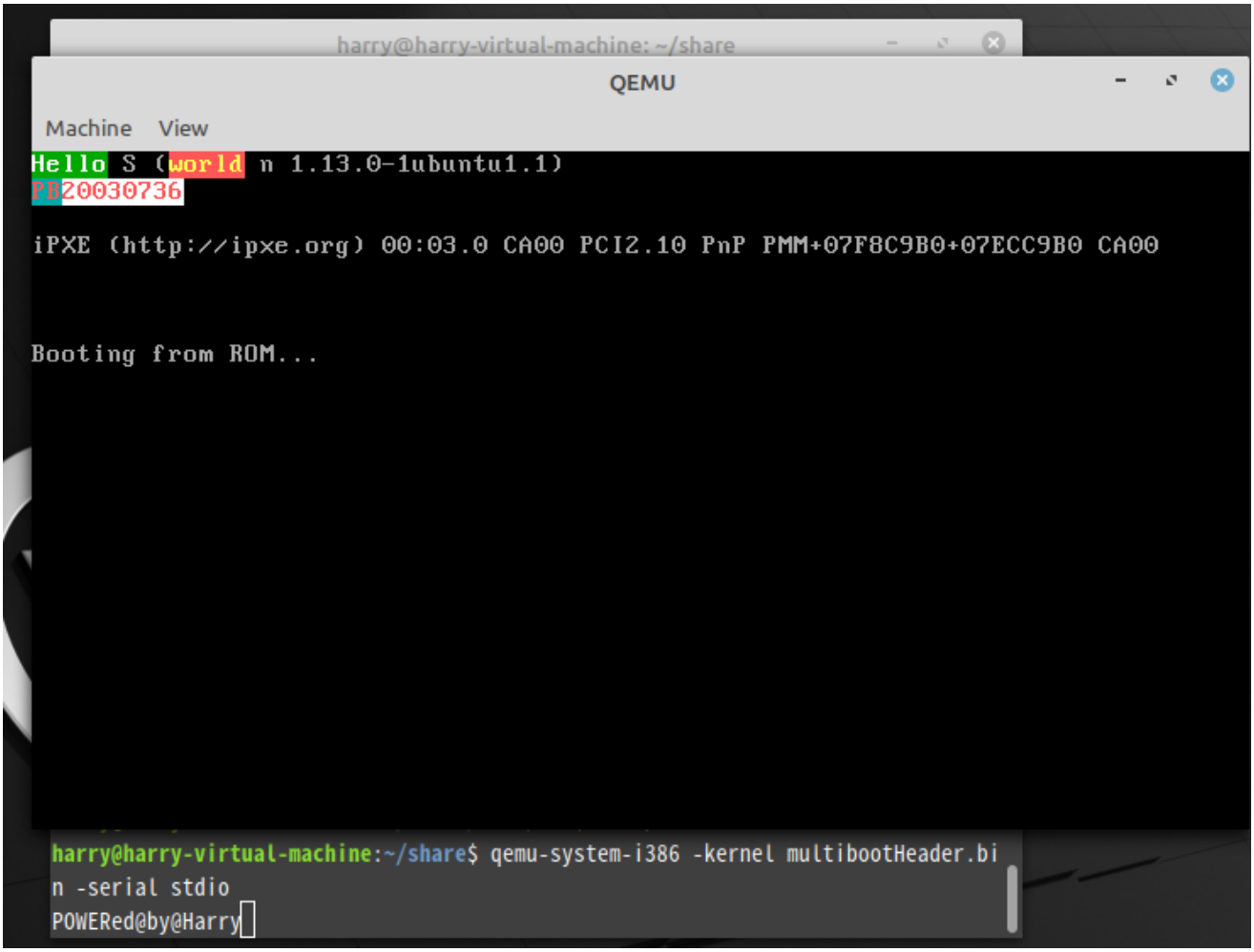
```
movb $0x50,%al
movw $0x3F8, %dx
outb %al, %dx
movb $0x4F,%al
movw $0x3F8, %dx
outb %al, %dx
movb $0x57,%al
movw $0x3F8, %dx
outb %al, %dx
movb $0x45,%al
movw $0x3F8, %dx
outb %al, %dx
movb $0x52,%al
movw $0x3F8, %dx
outb %al, %dx
movb $0x65,%al
movw $0x3F8, %dx
outb %al, %dx
movb $0x64,%al
movw $0x3F8, %dx
outb %al, %dx

# POWERed

movb $0x40,%al
movw $0x3F8, %dx
outb %al, %dx

#0x40 @

movb $0x62,%al
movw $0x3F8, %dx
outb %al, %dx
movb $0x79,%al
movw $0x3F8, %dx
outb %al, %dx
movb $0x40,%al
movw $0x3F8, %dx
outb %al, %dx
#0x40@
movb $0x48, %al
movw $0x3F8, %dx
outb %al, %dx
movb $0x61, %al
movw $0x3F8, %dx
outb %al, %dx
movb $0x72, %al
movw $0x3F8, %dx
```

```
outb %al, %dx
movb $0x72, %al
movw $0x3F8, %dx
outb %al, %dx
movb $0x79, %al
movw $0x3F8, %dx
outb %al, %dx

#串口:harry

nop
nop
hlt
```

编译过程

> *make* 过程 将汇编文件 *.s*转为可重定向文件 *.o*由 *ld*处理
>
> 对*ld*文件的解释 *ld -T* 使用自己编写的*linker script* 代替 *default*
>
> *. location number* 在*1M*处开始 将原汇编文件定义的文本段和*header*段 合并成新的文本段

运行结果


harry@harry-virtual-machine: ~/share
QEMU
Machine View
Hello S (world n 1.13.0-1ubuntu1.1)
PB20030736
iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+07F8C9B0+07ECC9B0 CA00
Booting from ROM...
harry@harry-virtual-machine:~/share$ qemu-system-i386 -kernel multibootHeader.bi
n -serial stdio
POWERed@by@Harry