

os 第一次实验报告

PB20000024 陈奕衡

一、实验原理

本实验使用 multiboot 启动协议，并且利用 qemu 虚拟机执行文件并呈现结果，成功实现 vga 输出与端口输出。

multiboot 启动协议不依赖于任何操作系统，也不关心引导程序的细节，但规范了引导程序与操作系统之间的接口，从而保证了操作系统的共存。此实验中 multiboot 启动协议是用于帮助我们直接进入 cpu 的保护模式以访问内存。

qemu 是一种通用的开源计算机仿真器和虚拟机，可执行 x86 等多种架构的处理器，支持 multiboot 协议运行实验所搭建的内核。本实验使用 x86 架构进行搭建。

vga 输出是利用显卡接口通过系统显存输出相应内容，本实验所提供的屏幕端口起始地址为 0xB8000。

uart 串口输出是通过端口将要输出的内容显示在终端上，本实验所提供的端口地址为 0x3F8。

二、源代码说明

代码分为三个部分，必要参数、串口输出以及 VGA 输出。

首先是定义协议中必要的三个必要参数，分别为 magic, flag 以及 checksum。其中 Magic 指向内核部分设为，无特殊要求因此 flag 取 0，之后设置 checksum 使得三者和为 0。

```

MAGIC_ITEM_NAME = 0x1badb002
FLAGS_ITEM_NAME = 0
CHECKSUM_ITEM_NAME = 0xe4524ffe
.section ".multibootHeader"      #先起一个section的名字
.align 4
.long MAGIC_ITEM_NAME
.long FLAGS_ITEM_NAME
.long CHECKSUM_ITEM_NAME
.text
.code32

```

之后进入代码段部分。串口输出部分使用了 0x3F8 端口，重复利用 al 寄存器从而输出 ok。

```

movb $0x4f, %al /* 输出字母o */
movw $0x3f8, %dx
outb %al, %dx /* put char */
movb $0x4b, %al #输出字母k
outb %al, %dx
nop

```

VGA 输出部分通过 movl 指令将内容 “pb20000024_cyh_helloworld_” 以绿底白字的形式输出。最后利用 hlt 指令让虚拟机停止。

```

movl $0x2f422f50,0xB8000 #VGA显存输出，起始位置0xB8000，结束位置0xB8034
movl $0x2f302f32,0xB8004
movl $0x2f302f30,0xB8008
movl $0x2f302f30,0xB800c
movl $0x2f342f32,0xB8010
movl $0x2f632f5f,0xB8014
movl $0x2f682f79,0xB8018
movl $0x2f482f5f,0xB801c
movl $0x2f6c2f65,0xB8020
movl $0x2f6f2f6c,0xB8024
movl $0x2f772f20,0xB8028
movl $0x2f722f6f,0xB802c
movl $0x2f642f6c,0xB8030
movl $0x2f5f2f21,0xB8034
nop
hlt

```

三、代码布局声明

代码段从物理内存 1M 位置后开始，以 multibootHeader 中参数声明开始占用 12 个字节，之后声明以 8 字节对齐，从而在 16 字节后开始保存 vga 输出以及串口输出的代码。

```

OUTPUT_FORMAT("elf32-i386", "elf32-i386", "elf32-i386")
OUTPUT_ARCH(i386)
ENTRY(start)

SECTIONS {
    . = 1M;
    .text : {
        *(.multibootHeader)
        . = ALIGN(8);
        *(.text)
    }
}

```

串口输出占用了 al 寄存器储存单次输出内容，并且占用 dx 寄存器保存端口地址。

VGA 输出，占用了 qemu 的显存地址，从 0xB8000 开始一直到 0xB802。其余指令未占用地址空间。

四、编译过程说明

利用 makefile 文件在终端输入 make 进行编译，得到相应的.o 文件以及可执行的.bin 文件，完成编译后就可可在终端利用 qemu 运行。

```

cyh@cyh-ROG-Strix-G732LWS-G732LWS:~/os/oslab1$ make
gcc -c -m32 --pipe -Wall -fasm -g -O1 -fno-stack-protector multibootHeader.S -o multibootHeader.o
ld -n -T multibootHeader.ld multibootHeader.o -o multibootHeader.bin

```

五、运行和运行结果说明

运行结果如上，VGA 显存部分成功输出内容，并且终端位置输出了相应的串口输出内容。

```

PB20000024_cyh_Hello world!_ntu1.1)

iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+07F8C9B0+07ECC9B0 CA00

Booting from ROM...

cyh@cyh-ROG-Strix-G732LWS-G732LWS:~/os/oslab1$ qemu-system-i386 -kernel multibootHeader.bin -serial stdio
OK

```

六、遇到的问题和解决方案说明

首先是串口输出内容的问题：如果使用 `movl` 指令直接给串口提交 2 个字符一次性输出，会导致串口只输出后一个字符。经过查找资料发现，串口大小为 16 位，因此一次只能输出一个字符。所以进行重复输出只能给 `al` 寄存器反复赋值，再利用 `out` 指令输出。

其次还有 vga 输出中利用 `movl` 指令一次输出两个字符时会倒序的情况，解决方案是将两个字符信息倒置。