Lab1 系统引导

俞星凯 171830635

实验目的

1.学习在 Linux 环境下编写、调试程序，初步掌握 Shell、 Vim、GCC、Binutils、Make、QEMU、GDB 的使用。
2.学习 AT&T 汇编程序的特点。

3.理解系统引导程序的含义，理解系统引导的启动过程。

实验内容

从实模式切换至保护模式，在保护模式下读取磁盘 1 号扇区中的 Hello World 程序至内存中的相应位置，跳转执行该 Hello World 程序，并在终端中打印 Hello, World!

实验过程

1.实模式切换保护模式
在系统启动时首先进入 8086 的实模式，但是由于实模式下的安全问题和其分段机制的问题，在 80386 中引入了保护模式，本次实验中模拟了 8086 实模式到 80386 保护模式的切换。

```
.global start
start:
        movw %cs, %ax
        movw %ax, %ds
        movw %ax, %es
        movw %ax, %ss
        cli # clear interuption
        inb $0x92, %al # Fast setup A20 Line with port 0x92, necessary or not?
        orb $0x02, %al
        outb %al, $0x92
        data32 addr32 lgdt gdtDesc # loading gdtr, data32, addr32
        movl %cr0, %eax
        orb $0x01, %al
        movl %eax, %cr0 # setting cr0
        data32 ljmp $0x08, $start32 # reload code segment selector and ljmp to
start32, data32
```

首先使用 cli 关闭中断，接着启动 A20 地址线，然后加载 GDTR 寄存器，接着设置 CR0 寄存器的 PE 位为 1 启动保护模式，最后长跳转切换至保护模式。
这里有两点值得注意：一是启动 A20 地址线的原因，如果 A20 地址线未开，那么地址的第 20 位永远为 0，导致地址空间不连续。二是 CS 的设置我们是通过长跳转指令实现的，因为 CS 不可直接修改。

2.保护模式准备工作

在保护模式下，首先初始化段寄存器组和栈帧，随后通过 jmp 去执行 boot.c 中的 bootmain 程序。

3.加载磁盘中的 Hello World 程序

在 boot.c 中使用 void readSect(void *dst, int offset)函数，通过 boot.h 中封装的读写接口，完成磁盘中 1 号扇区中 Hello World 程序的加载，并使用函数指针的方式执行该程序。

需要注意的是，将 elf 赋值为 0x8c00 的原因是在 app 的 Makefile 中可以看到其入口地址就是 0x8c00。

4.运行 Hello World 程序

Hello World 程序的实现采用了框架代码 start.S 中的实现办法，将 ah 设置为 0x0c，即黑底红字，通过循环控制将 al 不断的设置为需要打印的 ASCII 字符，然后使用 movw %ax, %gs:(%edi)来写入显存。

```
.code32

.global start
start:
        # TODO
        pushl $13
        pushl $message
        calll displayStr
loop32:
        jmp loop32

message:
        .string "Hello, World!\n\0"

displayStr:
        movl 4(%esp), %ebx
        movl 8(%esp), %ecx
        movl $((80*5+0)*2), %edi
        movb $0x0c, %ah
nextChar:
        movb (%ebx), %al
        movw %ax, %gs:(%edi)
        addl $2, %edi
        incl %ebx
        loopnz nextChar # loopnz decrease ecx by 1
        ret
```

实验收获

1.加深了对于系统引导启动过程的理解。

2.了解了段寄存器、GDTR、段表的格式。

3.初步掌握了 Linux 下编写和调试程序。