

《操作系统》实验报告

LAB1 系统引导

姓名：陈攀岭

学号：171860516

邮箱：171860516@smail.nju.edu.cn

一、实验要求

在保护模式下加载磁盘中 Hello World 程序运行。

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!”。

二、实验原理

1. 系统启动

系统启动时，计算机工作在实模式下，其中 CS:IP 指向 BIOS 的第一条指令，BIOS 将检查各部分硬件是否工作正常，然后按照 CMOS RAM 中设置的启动设备查找顺序来寻找可启动设备。

2. 实模式切换保护模式

关闭中断，打开 A20 数据总线，加载 GDTR，设置 CRO 的 PE 位（第 0 位）为 1b（表示进入保护模式），通过长跳转设置 CS 进入保护模式，初始化 DS，ES，FS，GS，SS，栈顶指针 ESP。跳转至 bootMain 函数。

3. 加载磁盘中的程序

中断关闭后，无法通过陷入磁盘中断调用 BIOS 进行磁盘读取。实验中提供的代码框架中实现了 readSec(void *dst, int offset) 这一接口（定义于 bootloader/boot.c 中），其通过读写（in, out 指令）磁盘的相应端口（Port）来实现磁盘特定扇区的读取。

通过上述接口读取磁盘 MBR 之后扇区中的程序至内存的特定位置并跳转执行。

三、实验步骤

1. 设置“utils/genboot.pl”权限

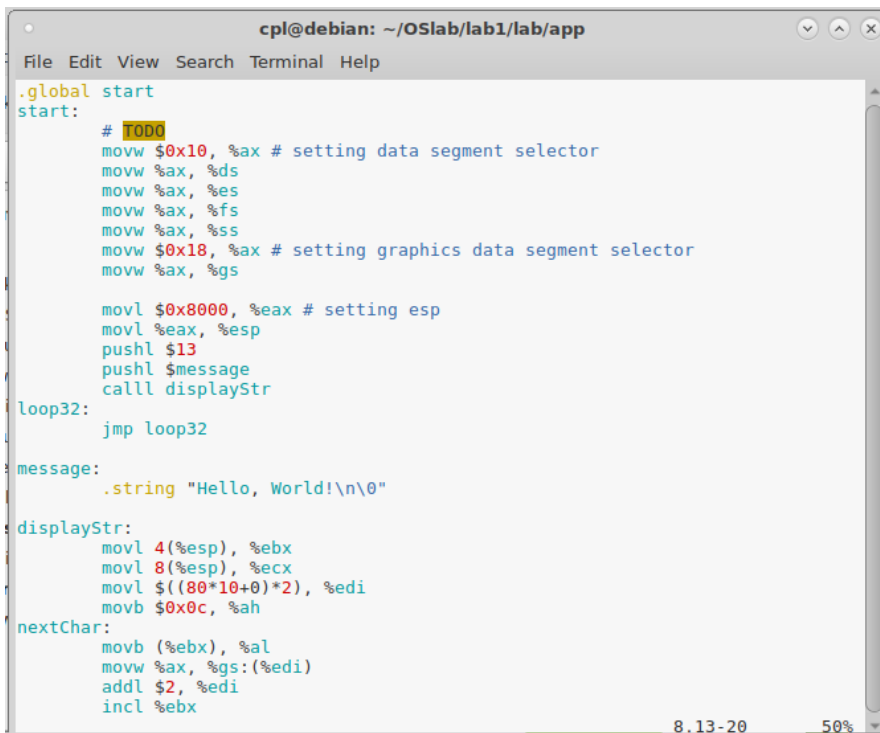
```
cpl@debian:~/OSlab/lab1/lab$ cd utils/  
cpl@debian:~/OSlab/lab1/lab/utils$ chmod 777 genboot.pl
```

2. 修改“bootloader/start.s”加载磁盘程序，在保护模式下跳转至“boot.c”中定义的函数“bootMain”。在 bootMain 函数中加载 Hello World 程序。



```
cpl@debian: ~/OSlab/lab1/lab/bootloader  
File Edit View Search Terminal Help  
  
movw %ax, %ss  
movw $0x18, %ax # setting graphics data segment selector  
movw %ax, %gs  
  
movl $0x8000, %eax # setting esp  
movl %eax, %esp  
jmp bootMain # jmp to bootMain in boot.c  
pushl $13  
pushl $message  
calll displayStr  
loop32:  
    jmp loop32  
  
message:  
    .string "Hello, World!\n\n"  
  
displayStr:  
    movl 4(%esp), %ebx  
    movl 8(%esp), %ecx  
    movl $((80*5+0)*2), %edi  
    movb $0x0c, %ah  
nextChar:  
    movb (%ebx), %al  
-- INSERT --  
64,6-9 71%
```

3. 修改 “app/app.s”，使得终端打印 “Hello World!”。



```
cpl@debian: ~/OSlab/lab1/lab/app
File Edit View Search Terminal Help
.global start
start:
    # TODO
    movw $0x10, %ax # setting data segment selector
    movw %ax, %ds
    movw %ax, %es
    movw %ax, %fs
    movw %ax, %ss
    movw $0x18, %ax # setting graphics data segment selector
    movw %ax, %gs

    movl $0x8000, %eax # setting esp
    movl %eax, %esp
    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*10+0)*2), %edi
    movb $0x0c, %ah
nextChar:
    movb (%ebx), %al
    movw %ax, %gs:(%edi)
    addl $2, %edi
    incl %ebx
```

4. make; make play, 终端打印出 “Hello World!”。



```
QEMU
SeaBIOS (version 1.10.2-1)

iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+07F90DD0+07ED0DD0 CA00

Booting from Hard Disk...

Hello, World!
```