自己动手写操作系统笔记

本文的编译的codes在IA32平台上

ubuntu14.04安装环境bochs，按照这个文章可以成功安装bochs，记得安装nasm

<http://blog.csdn.net/time4/article/details/25079417>

第一章

软盘模拟启动

当PC上电后，如果是从软盘启动，计算机就会检查软盘的0面0磁道1扇区，sector[510]=0x55，且sector[511]=0xAA结束，则BIOS认为它是一个引导扇区，也就是我们说的Boot Sector。当然一个正确的Boot Sector除了以0xAA55结束之外，还应该包含一段少于512B的执行码。

一旦BIOS发现了Boot Sector，就会将这512B的内容装在到内存的0000:7c00处(所以后面的代码开始会定义在0x7c00位置)，然后跳转到0000:7c00处将控制权彻底交给这段引导代码。到此为止，计算机不再由BIOS中固有的程序来控制，而变成操作系统的一部分来控制。

代码

chapter1/a/boot.asm

org 07c00h ; 告诉编译器程序加载到7c00处

mov ax, cs ; ds es两个段寄存器指向与cs相同的段，以便在以后进行数据操作的时候能定位到正确的位置

mov ds, ax

mov es, ax

call DispStr ; 调用显示字符串例程

jmp $ ; 无限循环

DispStr:

mov ax, BootMessage

mov bp, ax ; ES:BP = 串地址

mov cx, 16 ; CX = 串长度

mov ax, 01301h ; AH = 13, AL = 01h

mov bx, 000ch ; 页号为0(BH = 0) 黑底红字(BL = 0Ch,高亮)

mov dl, 0

int 10h ; 10h 号中断

ret

BootMessage: db "Hello, OS world!"

times 510-($-$$) db 0 ; 填充剩下的空间，使生成的二进制代码恰好为512字节

dw 0xaa55 ; 结束标志

(1)在NASM中，任何不被方括号[]括起来的标签或变量名都被认为是地址

foo dw 1

mov ax, foo 把foo的地址传给ax

mov bx, [foo] 把bx的值赋为1

(2)关于$和$$，$表示当前行被汇编的地址，$$表示程序被编译后的开始地址

ndisasm -o 0x7c00 boot.bin >> disboot.asm

00007C09 EBFE jmp short 0x7c09

$ = 0x7c09

$$ = 0x7c00

times 510-($-$$) db 0 ; 重复510-($-$$)遍，填充0x00,也就是说程序到这里有510B大小，加上结束标志正好512B

配置文件

~/MyOS/bochs-work/chapter1/a/bochsrc

命令

nasm boot.asm -o boot.bin

dd if=boot.bin of=a.img bs=512 count=1 conv=notrunc

