

# Lab1

---

## 任务1: Hello OS

### 1、创建 `boot.nasm` 文件

```
org 07c00h ; 告诉编译器程序加载到 7c00 处
mov ax, cs
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
dw 0xaa55 ; 结束标志
```

### 2、使用 `nasm` 汇编 `boot.asm` 生成"操作系统"的二进制代码

```
nasm boot.asm -o boot.bin
```

### 3、使用 `bximage` 命令生成虚拟软盘

```
bximage -> fd -> 1.44 -> a.img
```

#### 4、使用 `dd` 命令将操作系统写入软盘

```
dd if=boot.bin of=a.img bs=512 count=1 conv=notrunc
# if: 输入文件
# of: 输出设备
# bs: 代表一个扇区的大小
# count: 代表扇区数
# conv: 代表不作其他处理
```

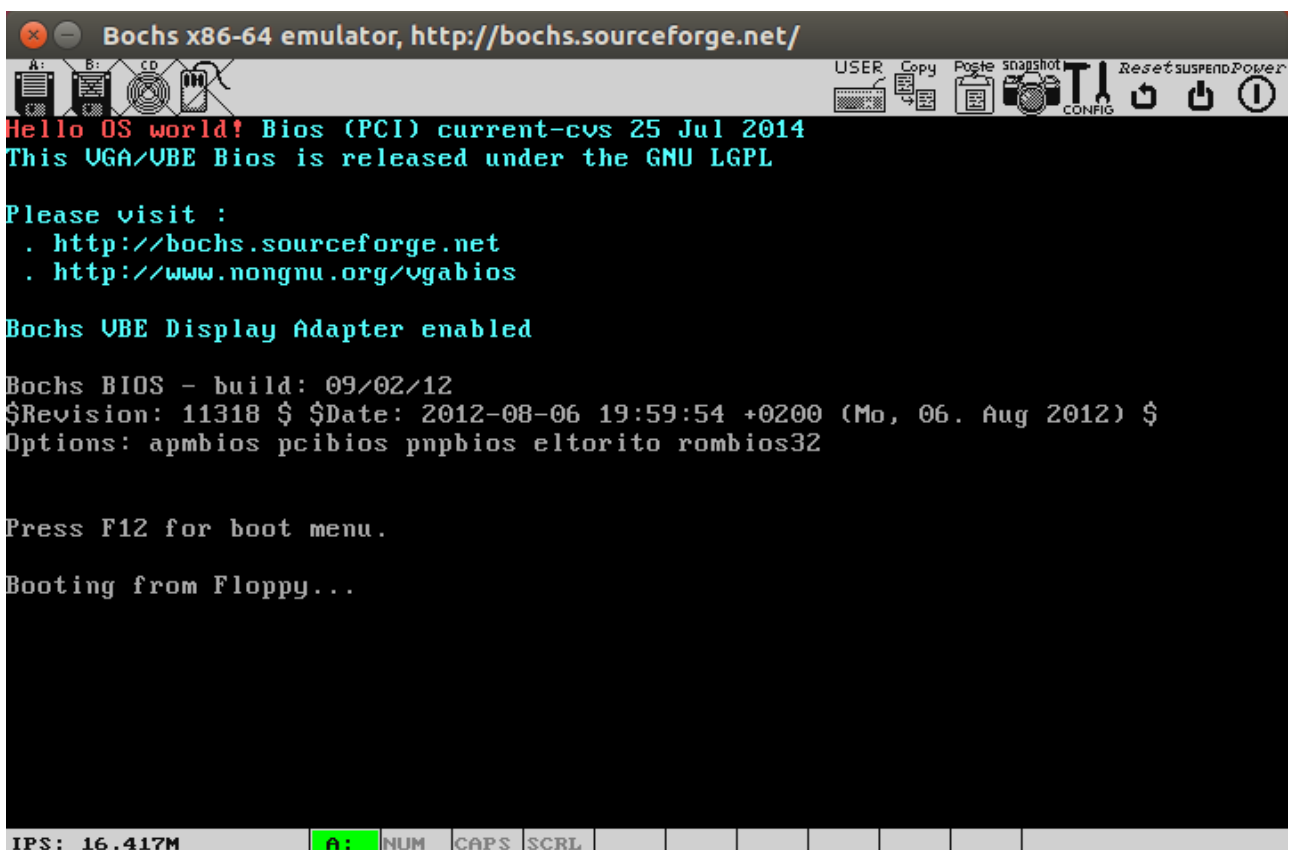
#### 5、配置 `bochsrc`

```
megs: 32 # 虚拟机内存大小(MB)
display_library: sdl # bochs使用的GUI库, 在Ubuntu下是sdl
floppya: 1_44=a.img, status=inserted # 虚拟机外设, 软盘为a.img文件
boot: floppy # 虚拟机启动方式, 从软盘启动
```

#### 6、启动 `bochs`

```
bochs -f bochsrc
```

按 `c` 即可显示如图



## 任务2：整数加法乘法

见代码