

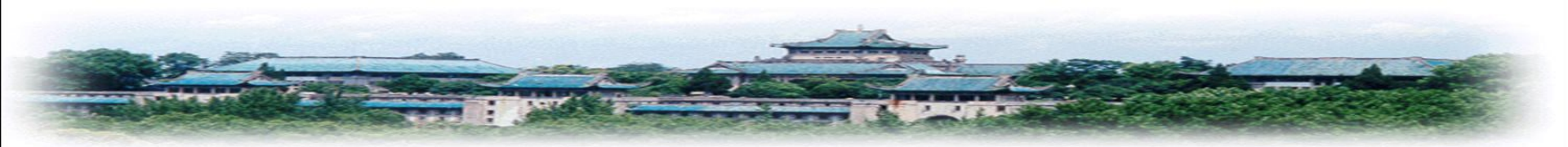
操作系统设计及实践

《操作系统原理》配套实验

操作系统课程组

2024年9月



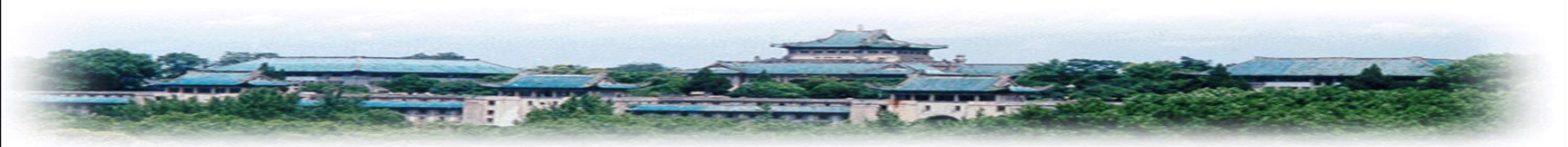


操作系统实验系列（一）

实验环境搭建：搭建实验的基本环境，熟悉开发与调试工具



武汉大学



一、实验目标

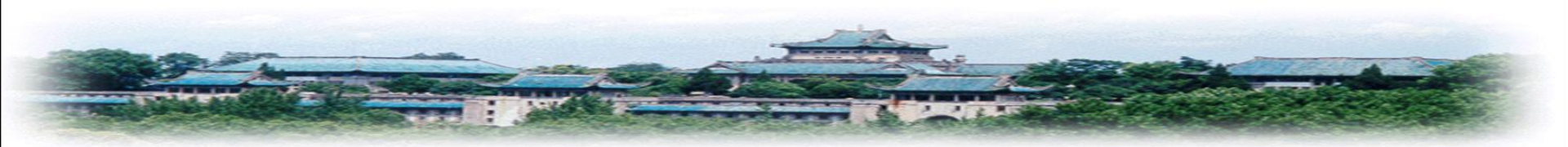
- 搭建基本实验环境，熟悉基本开发与调试工具
- 对应章节：第一、二章





- 汇编语言快速入门
 - http://docs.cs.up.ac.za/programming/asm/derick_tut/index.html
 - <http://heather.cs.ucdavis.edu/~matloff/50/LinuxAssembly.html>
 - <http://www.ibm.com/developerworks/cn/linux/l-assembly/>
 - <http://heather.cs.ucdavis.edu/~50/LinuxAssembly.html>
- 386编程基础
 - <https://pdos.csail.mit.edu/6.828/2008/readings/i386/toc.htm>
 - <http://faydoc.tripod.com/cpu/>





二、本次实验内容

1. 认真阅读章节资料
2. 在实验机上安装虚拟运行环境，并安装ubuntu（**实验室机器已安装，若需要可在自己笔记本电脑另行安装**）
3. 安装ubuntu开发环境，**32位环境**
4. **下载bochs源码**，编译并安装bochs环境
5. 使用bochs自带工具bximage创建虚拟软驱
6. 阅读、编译boot.asm，并反汇编阅读
7. 修改bochsrc，运行并调试你的第一个程序
8. 完成实验练习要求





1、安装环境注意事项

- Virtualbox及其增强包下载：
 - <https://www.virtualbox.org/wiki/Downloads>
- Ubuntu下载：32位Ubuntu（desktop-i386），14.04及以上
 - <http://mirrors.163.com/ubuntu-releases/>
- 修改ubuntu源方法：
 - <http://mirrors.163.com/.help/ubuntu.html>
- Bochs下载：bochs 2.7，注意要源码安装
 - <https://bochs.sourceforge.io/getcurrent.html>





1、安装环境注意事项

- Bochs编译注意事项

- 需要安装build-essential、libx11-dev、libxrandr-dev、libsdl1.2-dev、vgabios、bxiimage，可在编译过程中发现
- 需要设置配置参数./configure --各种参数
 - 见2.1.2节，并添加参数--with-sdl --enable-debugger --enable-disasm（在2.6以后不需要该项）

- 对下载的本书源码中第一个文件夹修改bochsrc中部分内容：

- <https://github.com/yyu/osfs00>
- 修改vgaromimage对应的文件位置，以你的实际安装位置为准
- 注释掉keyboard_mapping一行，或者做相应修改
- 增加display_library: sdl（某些linux下该项可选）



2、一段小代码

```
1      org      07c00h          ; 告诉编译器程序加载到7c00处
2      mov      ax, cs
3      mov      ds, ax
4      mov      es, ax
5      call     DispStr          ; 调用显示字符串例程
6      jmp      $                ; 无限循环
7  DispStr:
8      mov      ax, BootMessage
9      mov      bp, ax           ; ES:BP = 串地址
10     mov      cx, 16           ; CX = 串长度
11     mov      ax, 01301h       ; AH = 13, AL = 01h
12     mov      bx, 000ch        ; 页号为0 (BH = 0) 黑底红字 (BL = 0Ch, 高亮)
13     mov      dl, 0
14     int      10h              ; 10h 号中断
15     ret
16  BootMessage:                db      "Hello, _OS_world!"
17  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
18  dw          0xaa55           ; 结束标志
```





BIOS启动过程

- 当计算机加电后，一般会执行系统初始化软件
 - 完成基本IO初始化
 - 初始化硬件设备、建立系统的内存空间映射图→使得机器进入一个适合OS内核工作的状态
 - 引导加载功能
 - 引导加载程序把操作系统内核映像加载到RAM中，并将系统控制权传递给它。
- PC机：
 - 根据工业规范，计算机启动后，CPU会执行从一个特定地址开始执行系统初始化指令
 - PC中国化的初始化软件：BIOS / EFI
 - PC中NV上存储的软件，OS Boot Loader
- Intel 80386：
 - Step1: 计算机加电后，CPU从物理地址0xFFFFFFF0开始执行。
 - Step2: 在0xFFFFFFF0这里只是存放了一条跳转指令，通过跳转指令跳到BIOS例行程序起始点。
 - Step3: BIOS做完计算机硬件自检和初始化后，会选择一个启动设备（例如软盘、硬盘、光盘等），并且读取该设备的第一扇区(即主引导扇区或启动扇区)到内存一个特定的地址0x7c00处，然后CPU控制权会转移到那个地址继续执行。至此BIOS的初始化工作做完了，进一步的工作交给了OS的bootloader。





制作一个可启动的软盘

- 工具

- 编译源码

- >nasm boot.asm -o boot.bin (生成引导文件)
 - >nasm boot.asm -o boot.com (生成com文件)

- 产生一张虚拟软驱

- >bximage

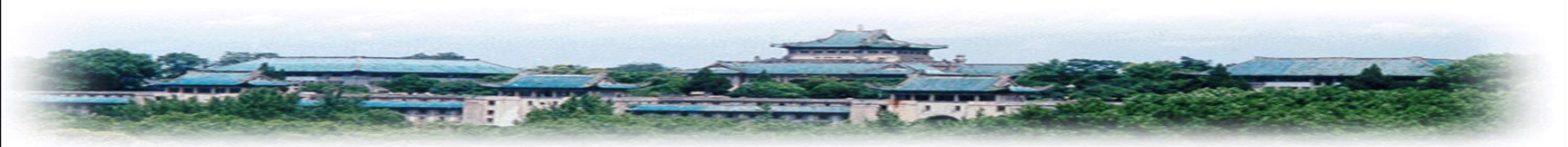
- 写引导盘

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

- 启动

- 修改bochsrc, bochs -f ./bochsrc 注意bochsrc不是bochs
源码目录下那个配置文件, 而是随书源码程序的目录





Bochs的调试基本命令

- 设置断点 b address
- 显示所有断点 info break
- 继续执行c
- 单步执行
 - s （可跳入函数）
 - n （跳过函数内部）
- 反汇编命令u 起始地址 终止地址
- 查看通用寄存器信息 r，段寄存器sreg，控制寄存器 creg

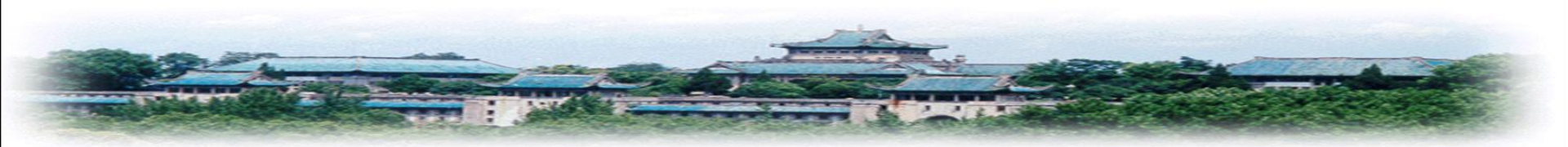




一些以前同学常出现的问题

- bochsrc配置文件到处乱放，应该在你的/home目录下，建立一个针对本实验的文件夹，然后把你的代码建立文件夹放进去。
- 不修改linux的应用库源，下载非常慢
- 把代码遗留在机房机器上，重启就什么也没有了，记得带走，也可以课后在自己笔记本上搭建一个全新环境。
- Virtualbox中共享文件夹不能访问问题，
 - 该目录的所有者是root，所属组是vboxsf
 - 把当前用户加到vboxsf组里即可：`sudo usermod -aG vboxsf $(whoami)`

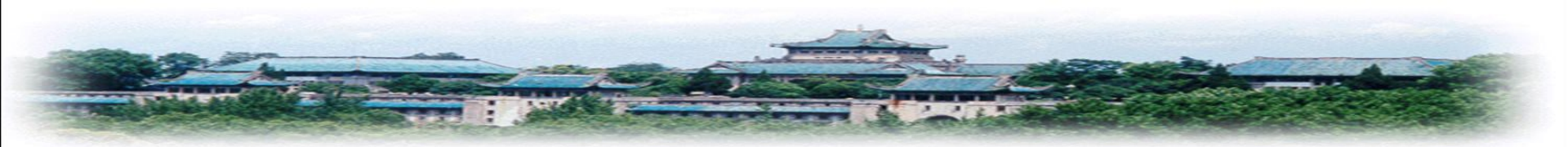




3.实验练习要求

1. 删除0xAA55，观察程序效果，找出原因
2. 修改程序中输出为，一个包含自己名字的字符串，调试程序
3. 把生成的可执行文件反汇编，看看输出的内容是怎样的，并在虚拟机启动过程，设置断点进行调试，在实验报告中截图
4. 为什么要jmp \$，如何改造程序，让这个输出过程执行100次
5. 回答：为什么要对段寄存器进行赋值
6. 回答：如何在该程序中调用系统中断





谢 谢！



武汉大学