软件综合实验之操作系统

把系统启动起来

陈香兰

中国科学技术大学计算机学院

July 1, 2016

提纲

实验准备

- 实验环境准备
  - 编译工具链：gcc、ld
  - 模拟环境：qemu-system-i386
- 基础知识和技能准备
  - i386实模式
  - 所使用的i386的启动协议
  - 通过直接写VGA显存来输出
  - 链接描述文件
  - 简单的编译、链接、映像制作脚本
  - 使用hexdump查看bin文件和img文件中的内容
    （对照：尝试查看你磁盘的MBR的内容）
  - 使用objdump反汇编目标文件“xxx.o”和elf文件
    （对照：尝试反汇编“xxx.bin”文件）

i386的分段机制

- I386体系结构采用分段机制
  - 逻辑地址=段：段内偏移
- 使用16位段寄存器来指明当前所使用的段
  - 有六个：cs，ss，ds，es，fs和gs
  - CPU规定了3个寄存器的专门的用途
    - cs 代码段寄存器，指向存放程序指令的段
    - ss 堆栈段寄存器，指向存放当前堆栈的段
    - ds 数据段寄存器，指向存放数据的段

I386的地址转换模式：实模式和保护模式

- 实模式（20位）
  - 16位段寄存器只记录段基址的高16位，
    因此段基址必须4位对齐（末4位为0）
  - 不采用虚拟地址空间，直接采用物理地址空间
  - 物理地址=段寄存器值*16+段内偏移
- 保护模式（32位）
  - 16位段寄存器无法直接记录段的信息，因此需要与全局描述符表GDT配合使用
  - GDT中记录了每个段的信息（段描述符），段寄存器只需记录段在GDT中的序号

本实验使用的i386的启动协议

- BIOS根据内置（可配置）的启动顺序，依次从潜在启动设备上搜索启动扇区
  - 网盘启动？
  - 磁盘启动？
  - 软盘启动？（本实验使用软盘启动）
    - 软盘的启动扇区最后两个字节应当是0xAA55
    - 软盘启动扇区的内容被加载到物理地址0x7C00处，然后跳转到这个地址上运行

VGA显存

- 本实验中字符界面规格：25行80列
- VGA显存的起始地址：0xB8000
- 每个字符需要2个字节：一个用于存放字符的ASCII码，一个用于存放该字符的显示属性

Attribute								Character							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

- 显示属性

Attribute							
7	6	5	4	3	2	1	0
Blink	Background color			Foreground color			

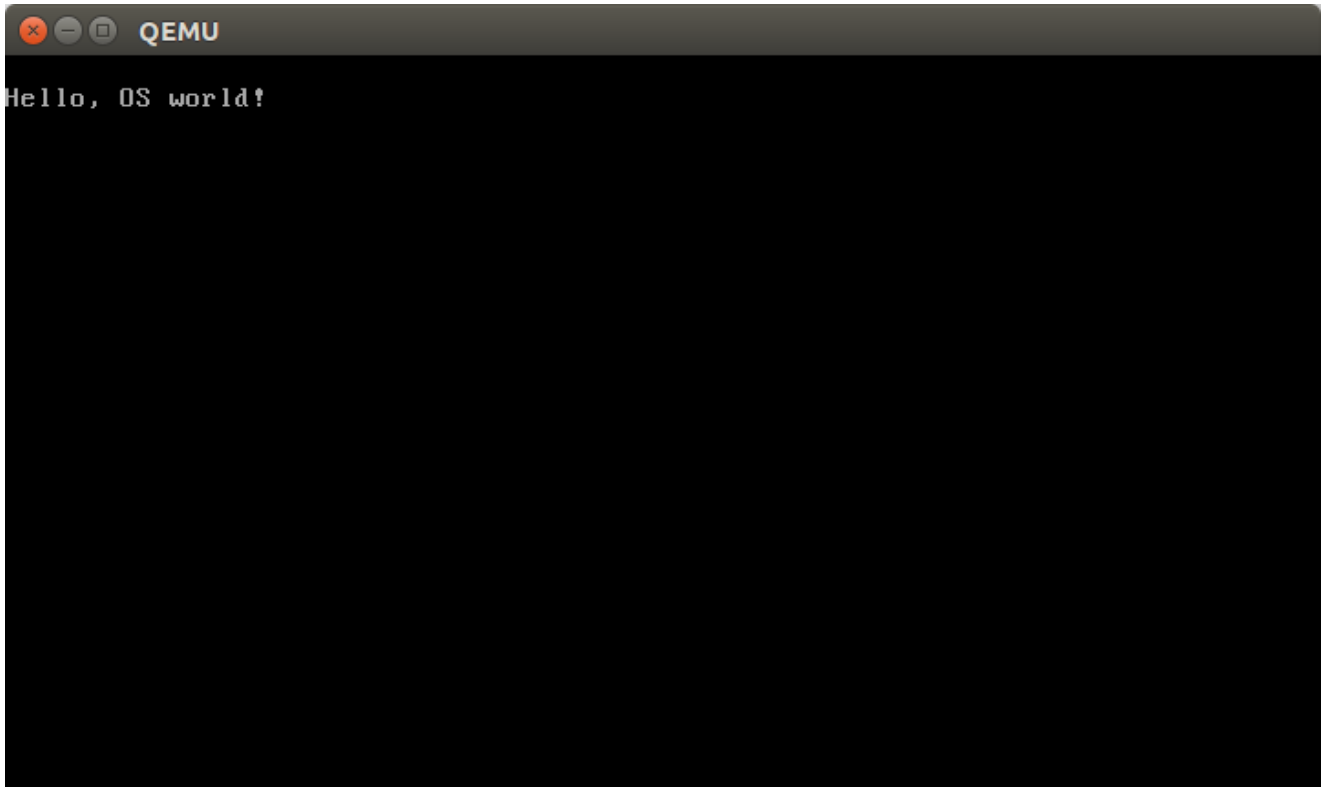
- 直接写VGA显存可以输出信息

实模式下的“Hello World!”

1. 阅读源代码
   - start16_hello.S
   - start16_hello.ld
2. 编译、链接、制作启动软盘的脚本
   1. 编译链接并制作成二进制映像
      - gcc -c -m32 start16_hello.S -o start16_hello.o
      - ld -T start16_hello.ld start16_hello.o -o start16_hello.elf
      - objcopy -O binary start16_hello.elf start16_hello.bin
   2. 制作启动软盘
      - dd if=/dev/zero of=a.img bs=512 count=2880
      - sudo losetup /dev/loop4 a_start16_hello.img
      - sudo dd if=start16_hello.bin of=/dev/loop4 bs=512 count=1

实模式下的“Hello World!”

3. 在qemu上启动
   - qemu-system-i386 -fa a_start16_hello.img



小结

谢谢！