




操作系统实验

Labs of Operating Systems

Lab 3 从实模式到保护模式

 **中山大学** 计算机学院（软件学院）
SUN YAT-SEN UNIVERSITY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

实验安排

实验-3：从实模式到保护模式

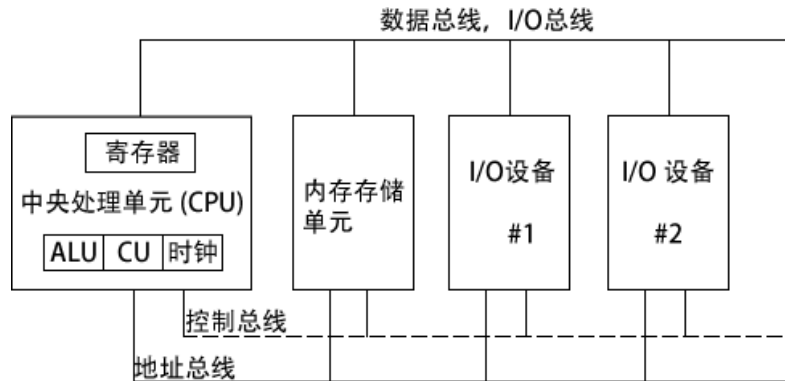
继续学习操作系统启动的原理，利用汇编语言实现OS启动到实模式（即20位地址空间），并切换到保护模式（即32位地址空间），在此基础上利用汇编或者C程序实现简单的应用

1. 回顾Lab2 32位汇编语言的基本语法、实模式下OS启动、简单的应用；
2. 实现从实模式到保护模式的转换；
3. 在保护模式下利用汇编/C/Rust等实现简单的应用；
4. 比较实模式和保护模式的不同；

更多详情请阅读gitee实验教程!

2

处理器架构



3

加载程序

在程序执行之前，需要用一种工具程序将其加载到内存，这种工具程序称为程序加载器 (program loader)。加载后，操作系统必须将 CPU 向程序的入口，即程序开始执行的地址。以下步骤是对这一过程的详细分解。

- 1) 操作系统 (OS) 在当前磁盘目录下搜索程序的文件名；
- 2) 如果程序文件被找到，OS 就访问磁盘目录中的程序文件基本信息，包括文件大小，及其在磁盘驱动器上的物理位置；
- 3) OS 确定内存中下一个可使用的位置，将程序文件加载到内存；
- 4) OS 开始执行程序的第一条机器指令（程序入口）。当程序开始执行后，就成为一个进程 (process)；
- 5) 进程自动运行。OS 的工作是追踪进程的执行，并响应系统资源的请求；
- 6) 进程结束后，就会从内存中移除；

4



IA-32处理器基本架构

x86 处理器有三个主要的操作模式：保护模式、实地址模式和系统管理模式；以及一个子模式：虚拟 8086 (virtual-8086) 模式，这是保护模式的特殊情况。

1) 保护模式 (Protected Mode)

保护模式是处理器的原生状态，在这种模式下，所有的指令和特性都是可用的。分配给程序的独立内存区域被称为段，而处理器会阻止程序使用自身段范围之外的内存。

2) 虚拟 8086 模式 (Virtual-8086 Mode)

保护模式下，处理器可以在一个安全环境中，直接执行实地址模式软件，如 MS-DOS 程序。换句话说，如果一个程序崩溃了或是试图向系统内存区域写数据，都不会影响到同一时间内执行的其他程序。现代操作系统可以同时执行多个独立虚拟 8086 会话。

3) 实地址模式 (Real-Address Mode)

实地址模式实现的是早期 Intel 处理器的编程环境，但是增加了一些其他的特性，如切换到其他模式的功能。当程序需要直接访问系统内存和硬件设备时，这种模式就很有用。

4) 系统管理模式 (System Management Mode)

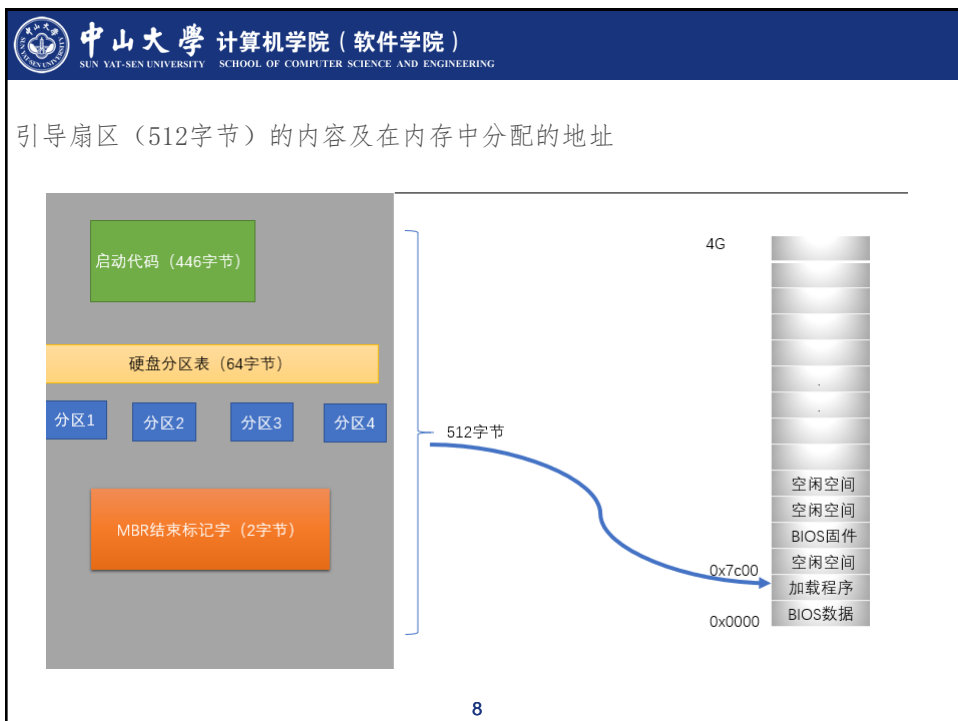
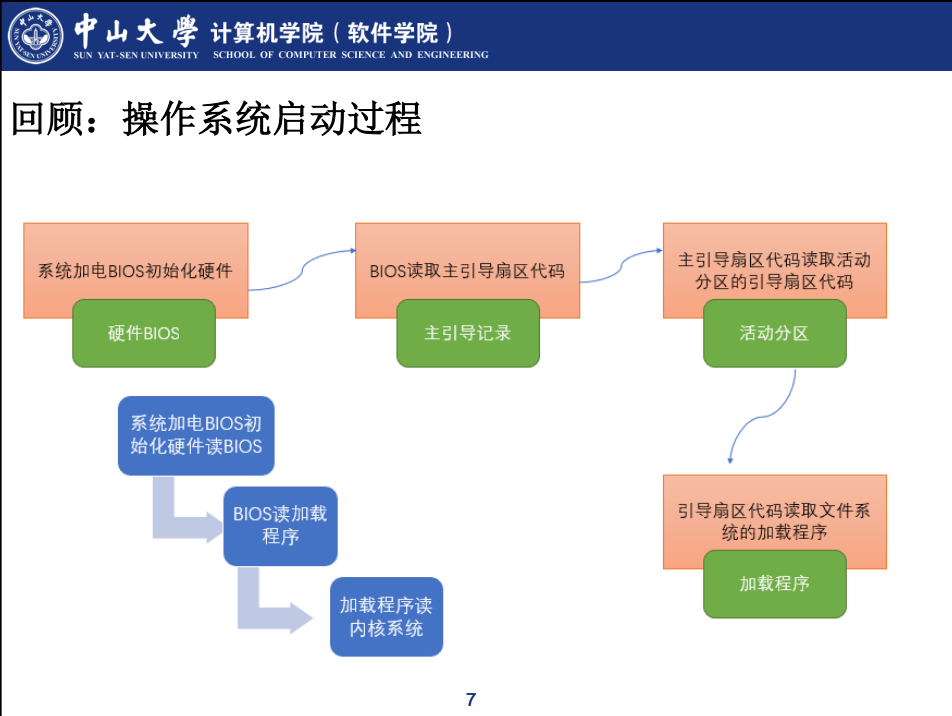
系统管理模式 (SMM) 向操作系统提供了实现诸如电源管理和系统安全等功能的机制。这些功能通常是由计算机制造商实现的，他们为了一个特定的系统设置而定制处理器。

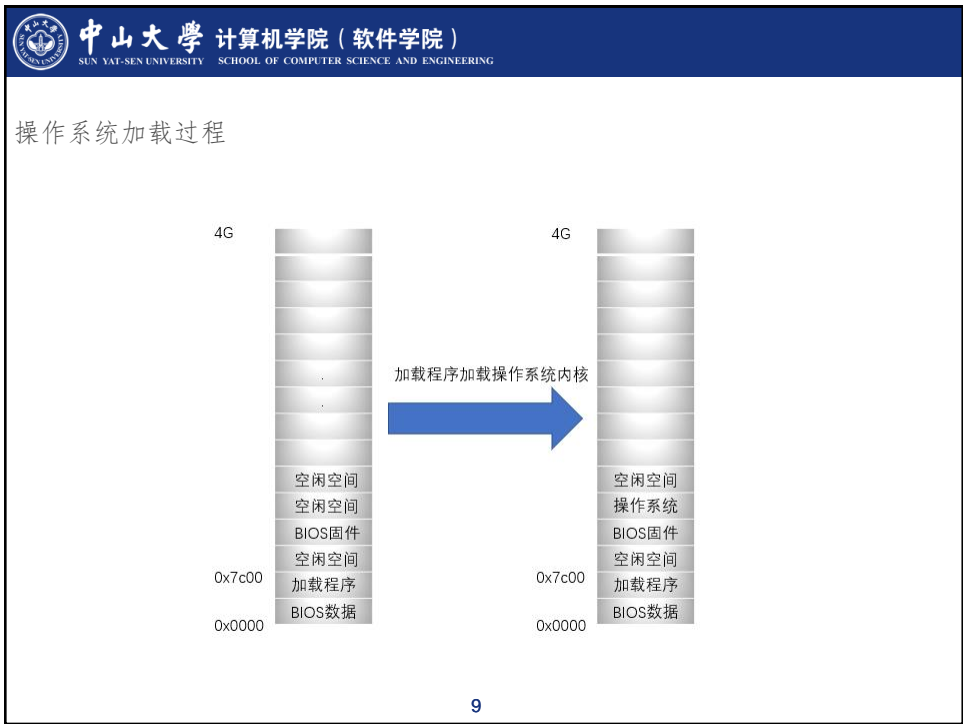


地址空间

- 在 32 位保护模式下，一个任务或程序最大可以寻址 4GB 的线性地址空间。从 P6 处理器开始，一种被称为扩展物理寻址 (extended physical addressing) 的技术使得可以被寻址的物理内存空间增加到 64GB。
- 在实地址模式下，IA-32处理器使用20位的地址线，可以访问220=1MB的内存，范围从0x0000到0xFFFF。但是，我们看到寄存器的访问模式只有32位，16位和8位，形如eax, ax, ah, al。那么我们如何才能使用16位的寄存器表示20位的地址空间呢？这在当时也给Intel工程师带来了极大的困扰，但是聪明的工程师想出来一种“段地址+偏移地址”的解决方案。段地址和偏移地址均为16位。此时，一个1MB中的地址，称为物理地址，按如下方式计算出来。

物理地址=(段地址<<4)+偏移地址





中山大学 计算机学院 (软件学院)
SUN YAT-SEN UNIVERSITY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

突破512字节的限制

在运行操作系统前，我们需要将操作系统内核程序从外存加载到内存中。显然，MBR无法胜任这项工作

1、通过BIOS中断 (int 13h) 实现磁盘内容读取；

```
19
20 disk_load:
21     push dx
22     mov ah, 0x02 ;BIOS读扇区功能
23     mov al, dh   ;读dh个扇区
24     mov ch, 0x00 ;选择柱面0
25     mov dh, 0x00 ;选择磁头0
26     mov cl, 0x02 ;读扇区2 (从boot_sector之后的第一个扇区)
27
28     int 0x13     ;BIOS interrupt
29     jc disk_error_flag
30     pop dx
31     cmp dh, al   ;al代表已读扇区数, dh是预期读扇区数
32     jne disk_error_count
33     ret
34 disk_error_flag:
35     mov bx, DISK_ERROR_FLAG_MSG
36     call print_string
37     jmp $
38 disk_error_count:
39     mov bx, DISK_ERROR_COUNT_MSG
40     call print_string
41     jmp $
```

展示

10



突破512字节的限制

2、通过LBA（Logical Block Addressing）方式读写磁盘；

- ✓ 硬盘是外围设备的一种，处理器和外围设备的交换是通过I/O端口进行的；
- ✓ 每一个端口在I/O电路中都会被统一编址。例如，主硬盘分配的端口地址是0x1f0~0x1f7；
- ✓ 因为端口是独立编址的，因此我们无法使用mov指令来对端口赋值，我们使用的是in,out指令；

```
in al, 0x21 ;表示从0x21端口读取一字节数据到al
in ax, 0x21 ;表示从端口地址0x21读取1字节数据到al, 从端口地址0x22读取1字节到ah
```

```
mov dx, 0x379
in al, dx ;从端口0x379读取1字节到al
```

```
; out指令
out 0x21, al ;将al的值写入0x21端口
out 0x21, ax ;将ax的值写入端口地址0x21开始的连续两个字节
mov dx, 0x378
out dx, ax ;将ah和al分别写入端口0x379和0x378
```

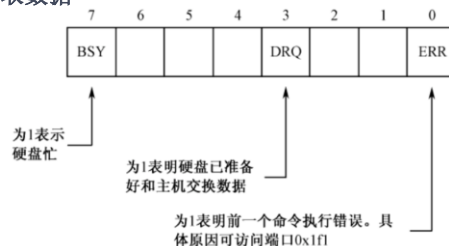
11




突破512字节的限制

2、通过LBA（Logical Block Addressing）方式读写磁盘；

- 设置起始的逻辑扇区号（LBA28）；
- 将要读取的扇区数量写入0x1f2端口；
- 向0x1f7端口写入0x20，请求硬盘读；
- 等待其他读写操作完成；
- 若在第4步中检测到其他操作已经完成，那么我们就可以正式从硬盘中读取数据



12



中山大学

SUN YAT-SEN UNIVERSITY

计算机学院（软件学院）

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

从实模式到保护模式

- 在保护模式下，所有的程序都会运行在自己的段中
- 段地址空间信息是通过段描述符(segment descriptor)来给出的；

31242322212019161514131211870


段基地址 31~24	G	D / B	L	A V L	段界限 19~16	P	DPL	S	TYPE	段基地址23~16
------------	---	-------	---	-------	-----------	---	-----	---	------	-----------

3116150

段基地址15~0	段界限15~0
----------	---------

- 保护模式下的段寄存器依然是 16 位，但其中保存的不再是段地址，而是段选择子；

13



中山大学

SUN YAT-SEN UNIVERSITY

计算机学院（软件学院）

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING


从实模式到保护模式

所有的段都会被保存在全局描述符表(GDT)中，实际上段选择子是全局描述符表的索引，类似数组访问array[i]中的i，但段选择子中还会包含其他信息，如下所示。

153210

描述符索引	T I	RPL
-------	--------	-----

14



中山大学

SUN YAT-SEN UNIVERSITY

计算机学院 (软件学院)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING


保护模式启动代码

Intel x86系列CPU有实模式和保护模式，实模式从8086开始就有，保护模式从80386开始引入。为了兼容，Intel x86系列CPU都支持实模式。现代操作系统都是运行在保护模式下（Intel x86系列CPU）。计算机启动时，默认的工作模式是实模式，为了让内核能运行在保护模式下，Bootloader需要从实模式切换到保护模式，切换步骤如下：

1. 准备好GDT(Global Descriptor Table)
2. 关中断
3. 加载GDT到GDTR寄存器
4. 开启A20，让CPU寻址大于1M
5. 开启CPU的保护模式，即把cr0寄存器第一个bit置1
6. 跳转到保护模式代码

GDT是Intel CPU保护模式运行的核心数据结构，所有保护模式操作的数据都从GDT表开始查找，[这里](#)有GDT的详细介绍。

15



中山大学

SUN YAT-SEN UNIVERSITY

计算机学院 (软件学院)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

GDT

GDT实际上是一个段描述符数组，保存在内存中。GDT的起始位置和大小由我们来确定，保存在寄存器GDTR中，GDTR的内容如下所示。

47

16 15

0

全局描述符表线性基地址	全局描述符表边界
-------------	----------

第21根地址线

在实模式下，第21根地址线的值恒为0,想进入保护模式时，首先需要打开第 21 根地址线；

```
in al, 0x92 ; 南桥芯片内的端口
or al, 0000_0010B
out 0x92, al ; 打开 A20
```

16



保护模式开关——CR0

CR0 是 32 位的寄存器，包含了一系列用于控制处理器操作模式和运行状态的标志位，其第0位是保护模式的开关位，称为PE (protect mode enable) 位。

```
cli          ; 保护模式下中断机制尚未建立，应禁止中断
mov eax, cr0
or  eax, 1
mov cr0, eax ; 设置 PE 位
```

17



参考资料

- [x86汇编\(Intel汇编\)入门](#)
- 《Intel汇编语言程序设计》第1-8章
- 《从实模式到保护模式》第1-8章
- <http://c.biancheng.net/makefile/>
- How to write a simple operating system
- The little book about OS development

18



中山大學 计算机学院 (软件学院)
SUN YAT-SEN UNIVERSITY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

谢谢

19