LISS
信息安全研究所

操作系统及安全设计

《操作系统及安全》配套实验

信安系操作系统课程组

2019年10月

操作系统设计实验系列（二）

认识保护模式（一）

武漢大学

一、实验目标

- 理解x86架构下的段式内存管理
- 掌握实模式和保护模式下段式寻址的组织方式、关键数据结构、代码组织方式
- 掌握实模式与保护模式的切换
- 掌握特权级的概念，以及不同特权之间的转移

二、本次实验内容

1. 认真阅读章节资料，掌握什么是保护模式，弄清关键数据结构：GDT、descriptor、selector、GDTR， 及其之间关系，阅读pm.inc文件中数据结构以及含义，写出对宏Descriptor的分析
2. 调试代码，/a/ 掌握从实模式到保护模式的基本方法，画出代码流程图，如果代码/a/中，第71行有dword前缀和没有前缀，编译出来的代码有区别么，为什么，请调试截图。
3. 调试代码，/b/，掌握GDT的构造与切换，从保护模式切换回实模式方法
4. 调试代码，/c/，掌握LDT切换
5. 调试代码，/d/掌握一致代码段、非一致代码段、数据段的权限访问规则，掌握CPL、DPL、RPL之间关系，以及段间切换的基本方法
6. 调试代码，/e/掌握利用调用门进行特权级变换的转移

三、实验解决问题与课后动手改

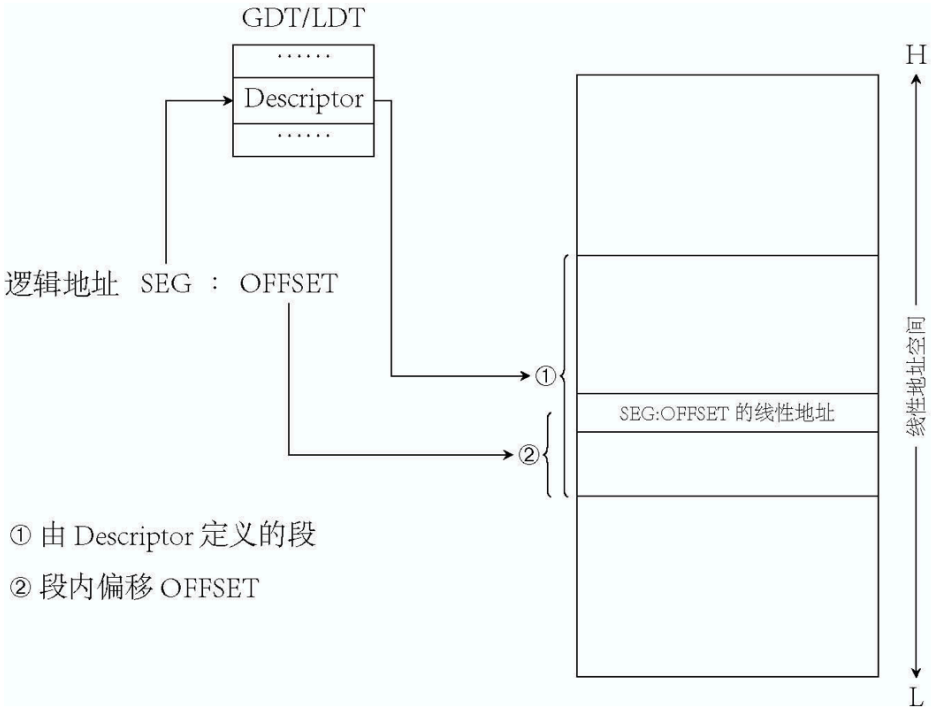
1. GDT、Descriptor、Selector、GDTR结构，及其含义是什么？他们的关联关系如何？pm.inc所定义的宏怎么使用？
2. 从实模式到保护模式，关键步骤有哪些？为什么要关中断？为什么要打开A20地址线？从保护模式切换回实模式，又需要哪些步骤？
3. 解释不同权限代码的切换原理，call, jmp，retf使用场景如何，能够互换吗？
4. 课后动手改：
    1. 自定义添加1个GDT代码段、1个LDT代码段，GDT段内要对一个内存数据结构写入一段字符串，然后LDT段内代码段功能为读取并打印该GDT的内容；
    2. 自定义2个GDT代码段A、B，分属于不同特权级，功能自定义，要求实现A–>B的跳转，以及B–>A的跳转。

四、需了解的知识

1. x86 CPU的基本模式：实模式、保护模式
   - 实模式
     - 地址总线宽度：20bit
     - 寄存器和数据总线宽度：16bit
     - 寻址空间是多少？
     - 实模式：PA=Segment*16+Offset

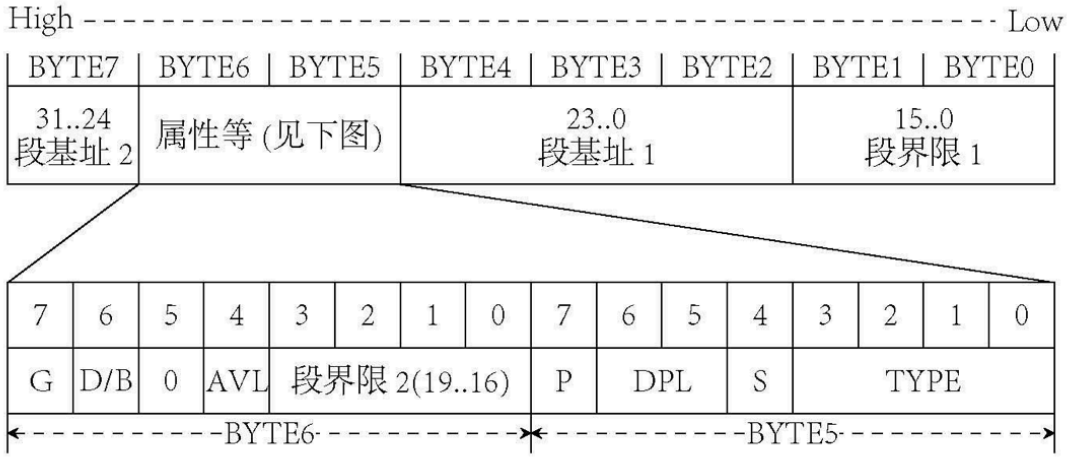
四、需了解的知识

1. x86 CPU的基本模式：实模式、保护模式
   - 保护模式
     - 段描述符
     - 选择子



四、需了解的知识

- 代码段、数据段段描述符



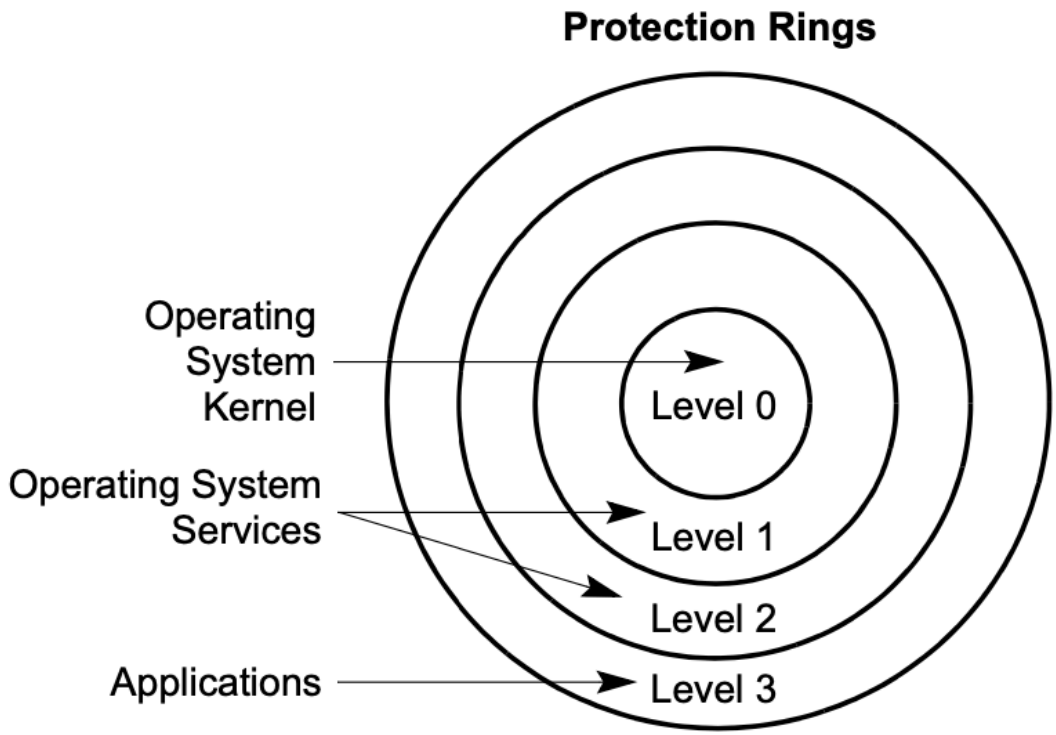
- 选择子

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
描述符索引													TI	RPL	

武漢大学

四、需了解的知识

2. 环保护



四、需了解的知识

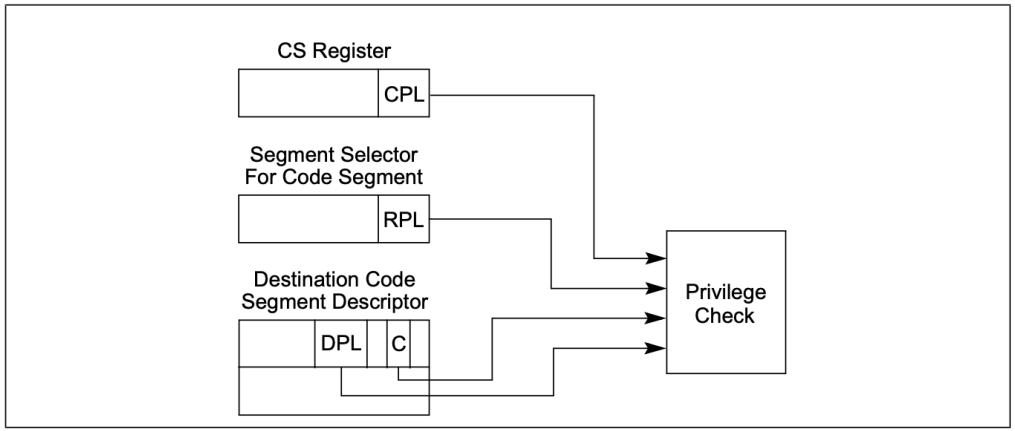
2. 环保护

- CPL
  - 描述当前执行程序或者任务的特权级。存储在CS和SS的bit 0，bit 1。当程序在不同特权级代码见转移，CPL会发生改变。
- DPL
  - 描述段或者门的特权级，存储在描述符的DPL字段中，是这个段的特权级别，用来表明访问这个段时候，所需要的特权。
- RPL
  - 描述选择子的特权级，段选择子的bit0和bit1，是可以重载的。例如：被调用的系统过程（CPL=0）从调用应用过程（CPL=3）收到一个选择子，就会把这个选择子的RPL设置成调用者的，从而表明当前是代表调用者的特权级在工作CPL=3，而不是CPL=0

四、需了解的知识

3. 段的特权类型

- 一致性代码段：
  - 允许低特权级的代码段（A），访问高特权级的代码段（B），即CPL-A>=DPL-B，此处RPL并不检查
  - 这里的高特权级代码不会访问敏感资源、也不会访问异常处理等系统代码，如某个纯粹的数学计算库
  - 一致性：当低特权级代码段，访问高特权级代码段时候，其CPL不发生变化，Why?
  - 不允许B代码段访问A，Why?



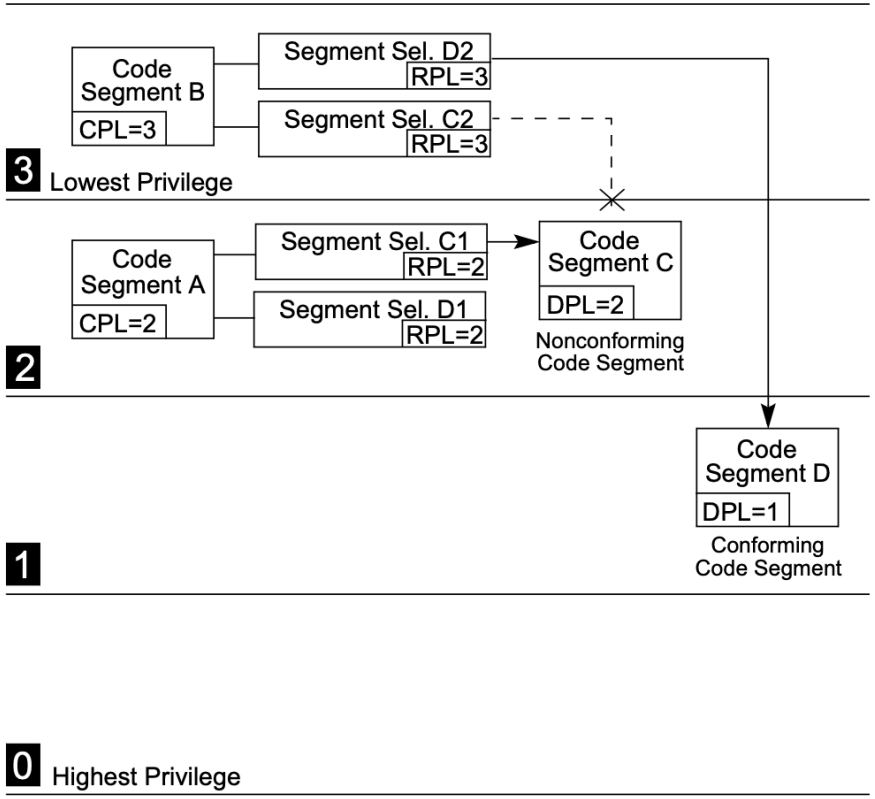
四、需了解的知识

3. 段的特权类型

- 非一致性代码段：
  - 只允许同特权级的代码能够访问，
  - 不允许不同级间访问:if A<>B, A不能访问B，B不能访问A
  - 作用：为了避免低特权级代码访问被操作系统保护起来的系统代码
  - CPL-A=DPL-B，RPL-A<=DPL-B(WHY RPL不能高于，因为RPL跟调用者有关)
- 数据段：
  - 高特权级别代码可以访问低特权级别数据
  - 同特权级别代码可以访问同特权级别数据
  - 不允许低特权级别代码访问高特权级别数据
  - 确保数据的完整性，避免被破坏

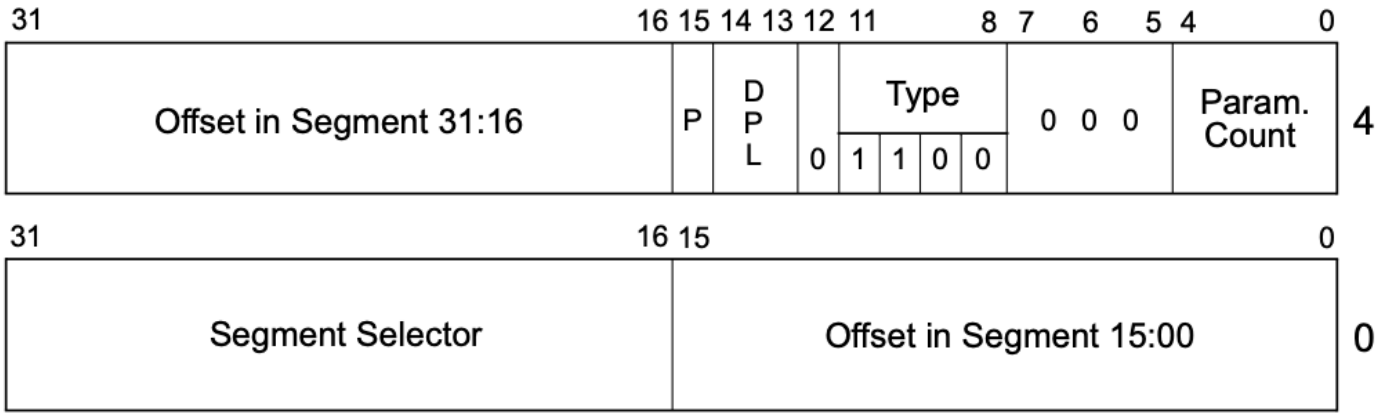
四、需了解的知识

3. 段的特权类型



四、需了解的知识

4. 不同特权级别段之间的代码转移
   - 一致代码段，直接访问，只能是CPL>=目标代码段的DPL
   - 非一致代码段，直接访问，只能同级CPL，且调用RPL<=目标DPL
   - 如何实现任意的呢？调用 CALL GATE



DPL Descriptor Privilege Level
P Gate Valid

四、需了解的知识

4. 不同特权级别段之间的代码转移
   - 本质上是一个添加了属性的特殊入口地址
   - 代码A→调用门G→代码B
   - CPL-A,RPL-A <= DPL_G
   - 如果B为一致代码段
     - CPL-A, RPL-A <= DPL_G
     - CPL-A>=DPL_B
     - 实现了从低特权代码A-->高特权代码B
   - 如果B为非一致代码段
     - CALL：CPL-A>=DPL_B
     - JMP: CPL-A = DPL_B

谢　谢！