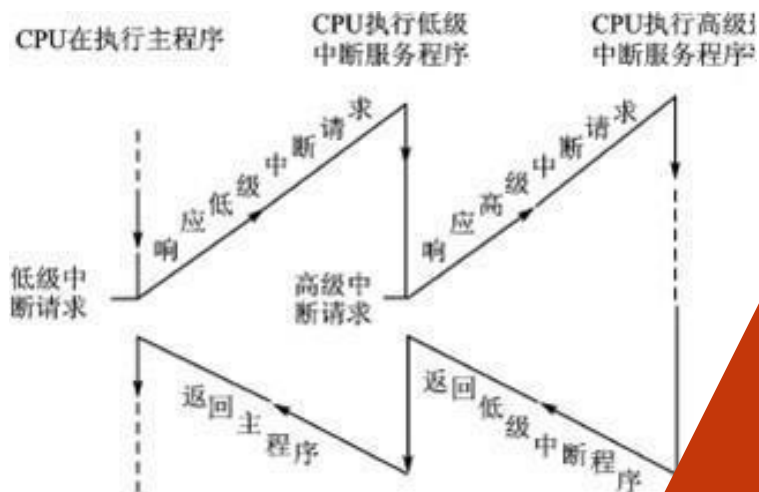




## 第二章：处理器管理



2.1 处理器状态

2.2 中断技术

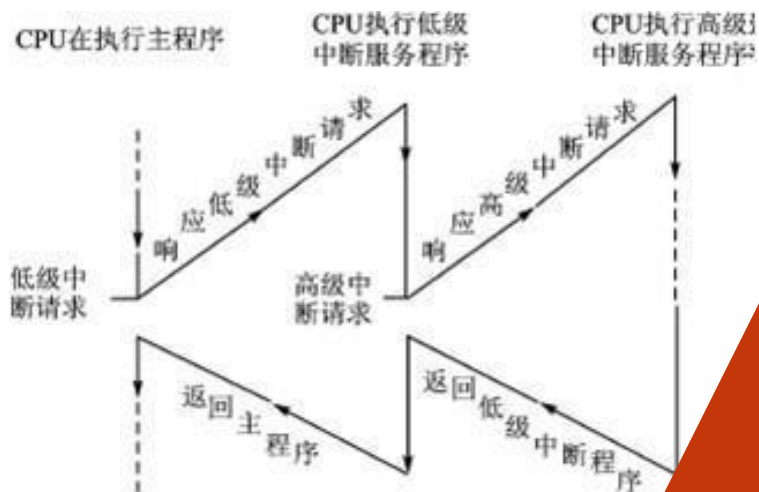
2.3 进程及其实现

2.4 线程及其实现

2.6 处理器调度



## 第二章：处理器管理



2.1 处理器状态

2.2 中断技术

2.3 进程及其实现

2.4 线程及其实现

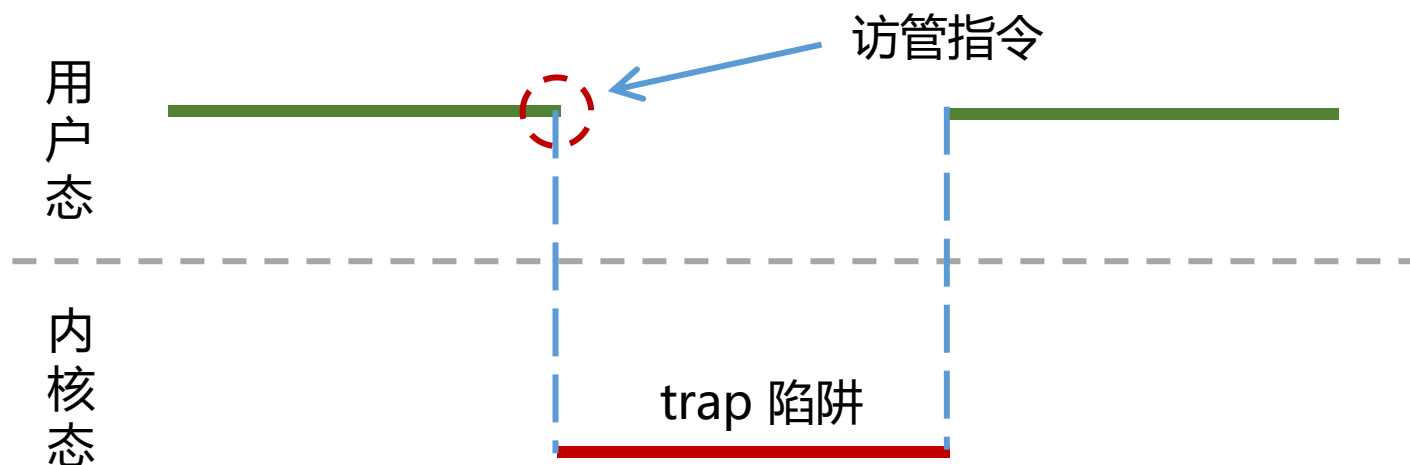
2.6 处理器调度



## 2.1 处理器状态

### 2.1.1 处理器

### 2.1.2 程序状态字





## 2.1.1 指令系统和寄存器

- 机器指令的集合称**指令系统**，反映了一台机器的功能和  
处理能力，分为以下六类：
  1. **数据处理**：执行算术和逻辑运算
  2. **转移**：改变指令执行序列，如无条件转移、条件转移等
  3. **数据传送**：用于在处理器的寄存器和寄存器、寄存器和存储器单元、存储器单元和存储器之间交换数据
  4. **移位**：算术、逻辑、循环移位；
  5. **字符串**：字符串的传送、比较、查询、转换
  6. **I/O类**：用于启动外围设备，让主存和外围设备之间交换数据



## 2.1.1 指令系统和寄存器

- 计算机系统的处理器包括一组寄存器
  - 寄存器个数根据处理器型号不同而异
  - 构成一级存储，比主存容量小，但访问速度快
  - 寄存器所存储的信息构成了处理器现场
- Intel x86处理器包含：

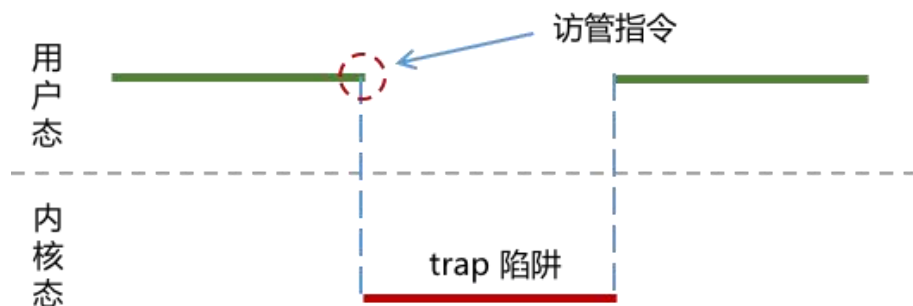
- ❑ 通用寄存器：EAX、EBX、ECX、EDX
- ❑ 指针及变址寄存器：ESP、EBP、ESI、EDI
- ❑ 段选择寄存器：CS、DS、SS、ES、FS、GS
- ❑ 指令指针寄存器和标志寄存器：EIP、EFLAGS
- ❑ 控制寄存器：CR0、CR1、CR2、CR3



## 2.1.1 指令系统和寄存器

- 在多道程序设计环境中，从资源管理和控制程序执行的角度出发，必须把指令系统中的指令分作两部分：

- **特权指令**：仅在内核态下才能执行的指令
- 非特权指令



- 只有操作系统才能执行指令系统中的全部指令（特权指令和非特权指令）
- 用户程序只能执行指令系统中的非特权指令



## 2.1.1 指令系统和寄存器

- 例，属于特权指令的有：

- 置程序状态字 PSW 指令
- 启动外围设备进行输入/输出
  - 只能在操作系统程序中执行，

否则会出现多个用户程序竞争使用外围设备而导致I/O混乱

- 设置时钟、清空内存、置中断屏蔽位、修改寄存器值等

如果用户程序试图执行特权指令，将会产生保护性中断，转交给操作系统的“用户非法执行特权指令”的特殊处理程序处理



## 2.1.1 指令系统和寄存器

- 根据程序对资源和机器指令的使用权限，处理器状态至少包括：
  - **内核态**：可执行全部机器指令，访问所有内存单元、系统资源，改变处理器状态
  - **用户态**：执行非特权指令，访问进程内地址单元，防止操作系统程序和其它用户程序受到侵害





## 2.1.1 指令系统和寄存器

- Intel x86的处理器状态：
  - 0级：操作系统内核级；处理I/O、存储管理和其他关键操作
  - 1级：系统调用处理程序级；通过执行系统调用，获得特定的和受保护的程序服务
  - 2级：共享库过程级：可以被多个运行进程共享，用户程序可以调用这些过程，读取它们的数据，但是不能修改它们
  - 3级：为用户程序级：受到的保护最少



## 2.1.1 指令系统和寄存器

- 处理器状态的转换

处理器做出响应并交换程序状态字，从而进行内核态

- 用户态向内核态转换

1. 使用访管指令，执行系统调用；
2. 运行过程中产生（其它进程的）中断事件，如I/O操作完成；
3. 程序运行过程中发生异常事件：除零操作、目态执行特权指令；

- 内核态向用户态转换

1. 由处理器指供一条加载程序状态字的特权指令



## 2.1.1 指令系统和寄存器

- 处理器状态的转换

- 用户态向内核态转换

1. 使用访管指令，执行系统调用；
2. 运行过程中产生（其它进程的）中断事件，如I/O操作完成；
3. 程序运行过程中发生异常事件：除零操作、目态执行特权指令；

这三种情况都是通过中断机构发生的，因此中断是目态到管态转换的唯一途径

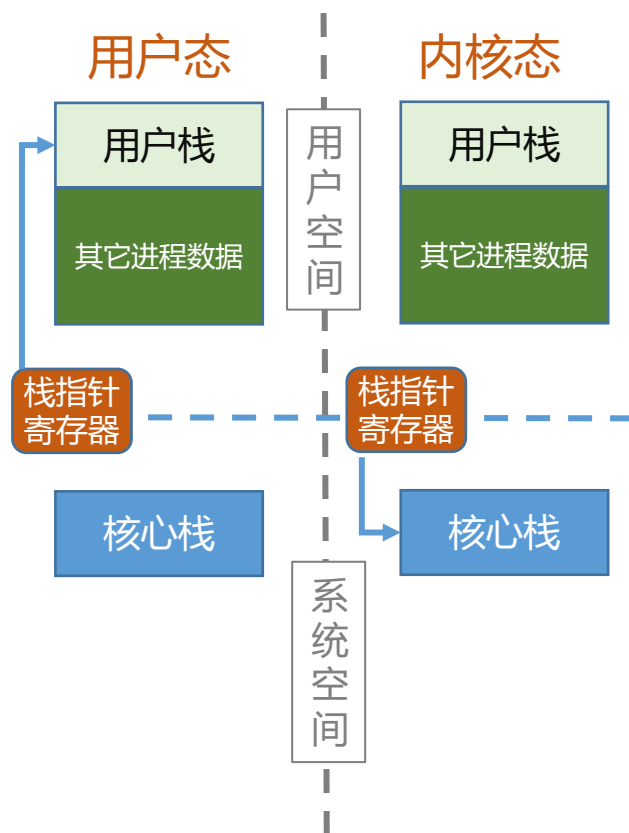
- 内核态向用户态转换

1. 由处理器提供一条加载程序状态字的特权指令



## 2.1.1 指令系统和寄存器

- 用户栈：用户进程空间中的内存区域
  - 进程中的局部变量、子函数参数、返回值、返回地址等
- 核心栈：操作系统空间中的一块区域
  - 嵌套的中断现场
  - 系统调用的参数、返回值、返回地址等





## 2.1.2 程序状态字

- 计算机如何知道当前处于何种工作状态，来决定是否可以执行特权指令？
  - 通常操作系统都引入程序状态字 PSW (Program Status Word) 来区别不同的处理器工作状态
- 每个正在执行的程序都有一个与其执行相关的PSW，而每个处理器都会设置一个程序状态字寄存器



## 2.1.2 程序状态字

- 程序状态字寄存器一般包括以下内容：
  - 程序基本状态：
    - 程序计数器：指明下一条执行的指令地址
    - 条件码：表示指令执行的结果状态
    - 处理器状态位：指明当前的处理器状态，如目态或管态、运行或等待
  - 中断码：保存程序执行时当前发生的中断事件
  - 中断屏蔽位：指明程序执行中发生中断事件时，是否响应出现的中断事件



## 2.1.2 程序状态字

- 大多数计算机的处理器现场中可能找不到一个称为程序状态字寄存器的具体寄存器，但总是有一组控制与状态寄存器实际上起到了这一作用
- Intel x86处理器中：
  - 由标志寄存器EFLAGS和指令指针寄存器EIP组成，均为32位