操作系统

第2章第2讲
中断/异常的响应和处理

文艳军（教授）
计算机学院

回顾

一. 版本1内核简介

两个进程，task0，task1， int 0x81

二. 中断：INTR，中断处理程序，断点和恢复点

三. 异常：中断号的分配，异常的类型

目录

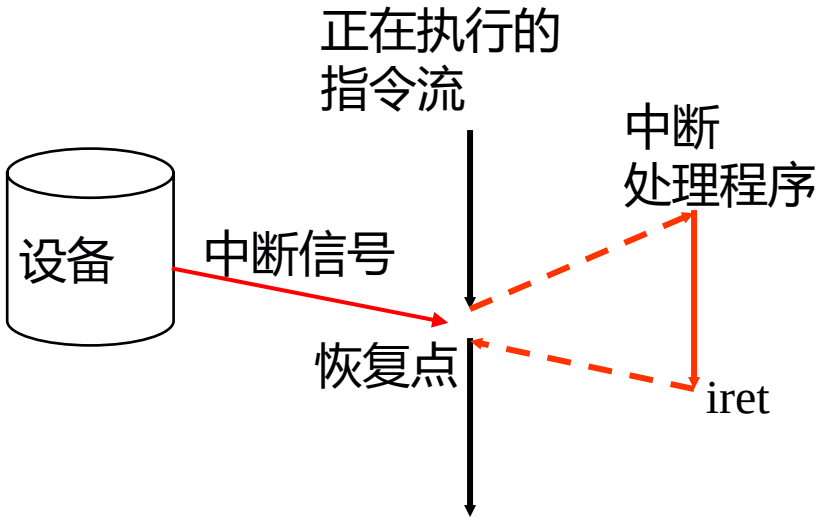
一. 中断/异常的处理过程

二. 演示：除零异常的响应

三. 独学&讨论

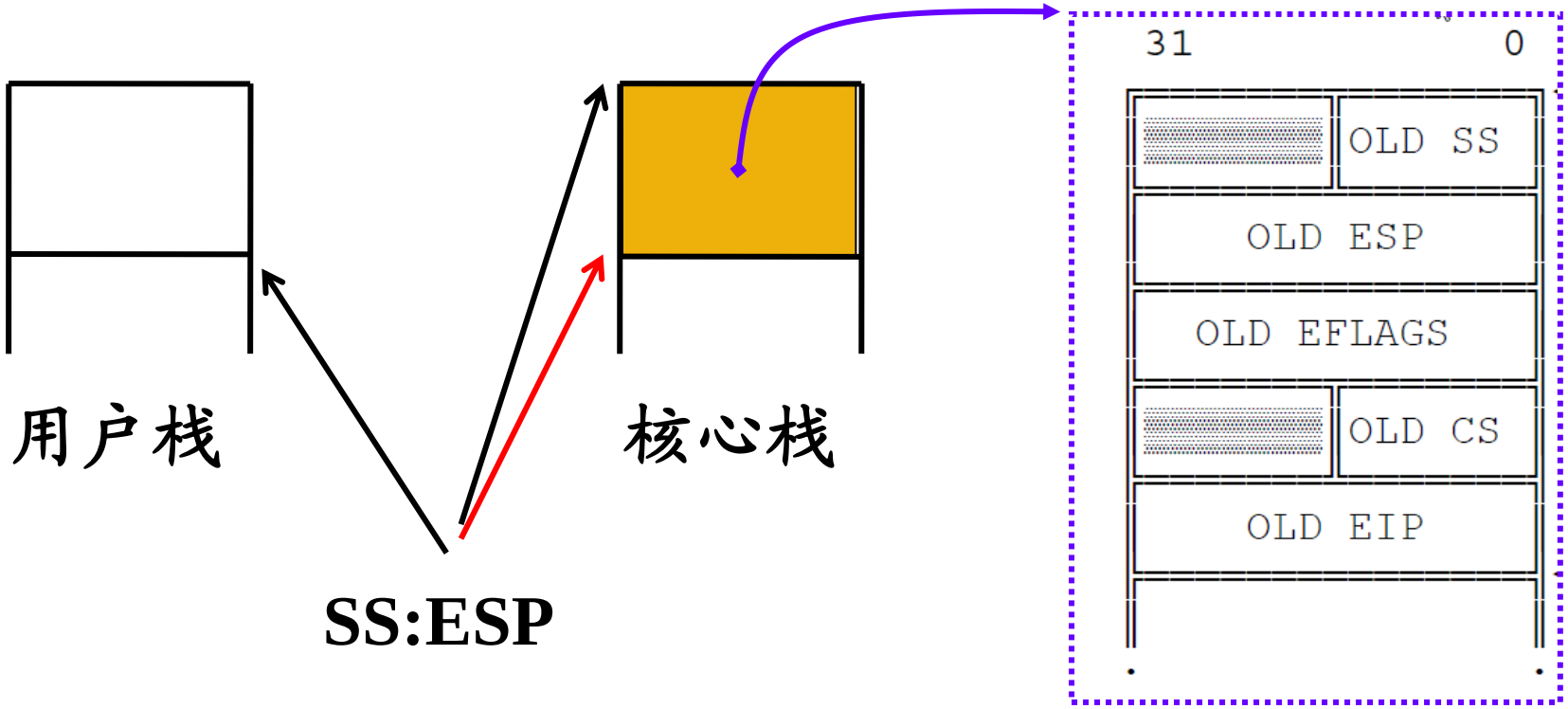
一. 中断/异常的处理过程

- 处理过程
  - CPU



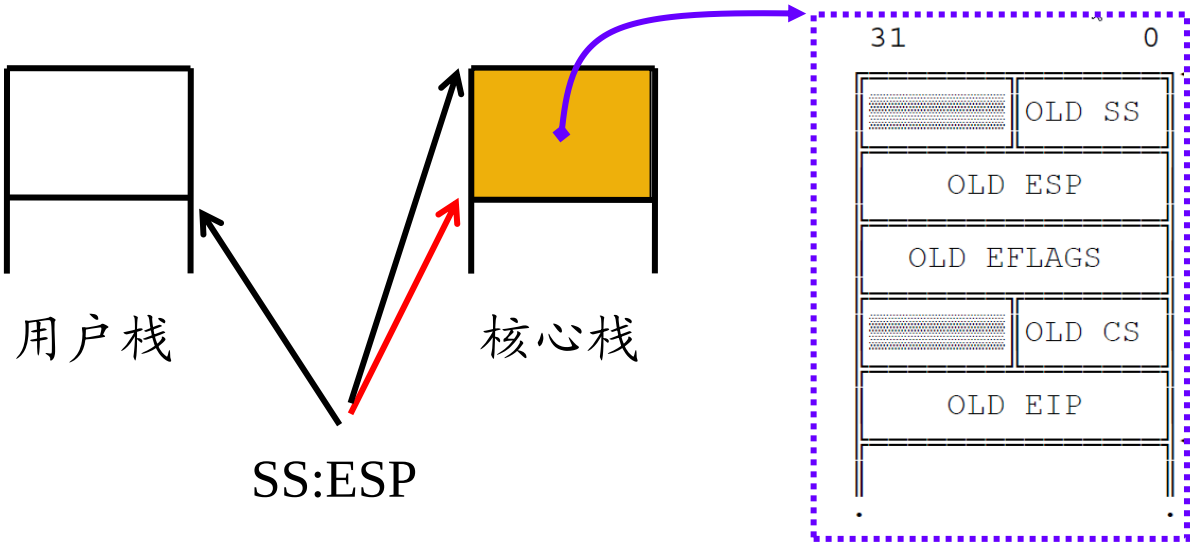
一. 中断/异常的处理过程

- 处理过程
  - CPU:
    a) 切换到核心栈，保存中断现场



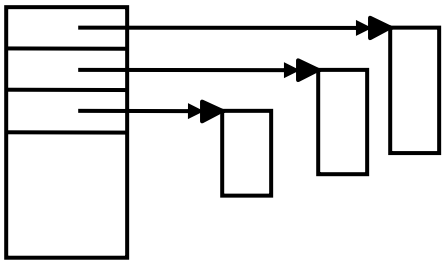
一. 中断/异常的处理过程

- 处理过程
  - CPU:
    a) 切换到核心栈，保存中断现场
    b) 查中断向量表，转到中断处理程序，同时切换到核心态



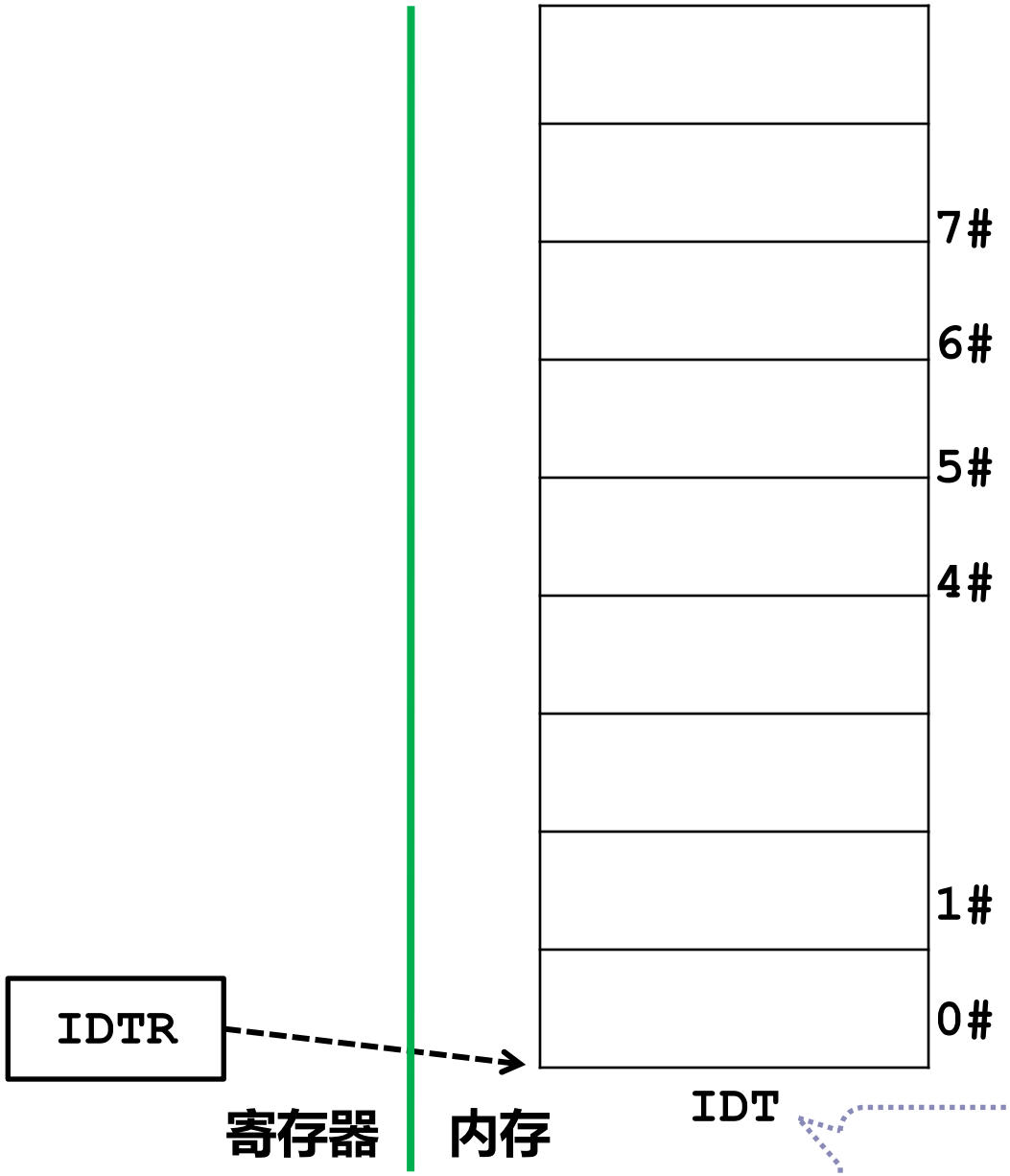
中断向量表

中断向量表:一片存放中断处理程序入口地址的内存单元。



中断向量由操作系统初始化程序进行设置，硬件按中断号的不同通过中断向量表跳转到相应处理程序。

80×86 的中断向量表



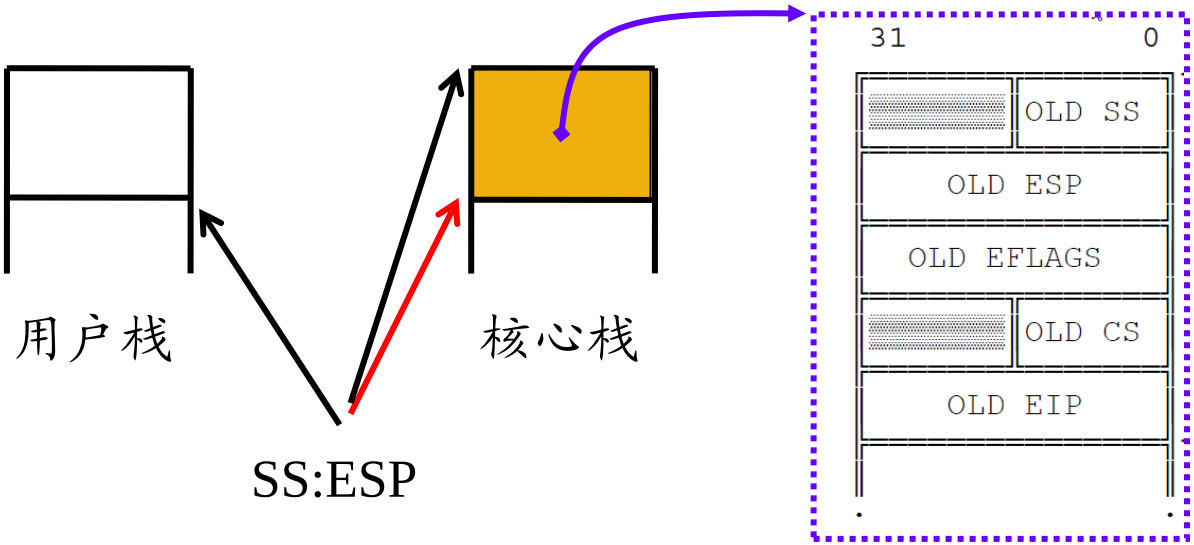
中断描述符表
Interrupt Descriptor Table

中断向量表

位置	子位置	函数入口地址
[54c0, 5cc0)	idt	
	idt[0x20]	**timer_interrupt**
	idt[0x28]	parallel_interrupt
	idt[0x2d]	irq13
	idt[0x80]	**system_call**
	idt[0x81]	display_interrupt

一. 中断/异常的处理过程

- 处理过程
  - CPU:
    a) 切换到核心栈，保存中断现场
    b) 查中断向量表，转到中断处理程序，同时切换到核心态

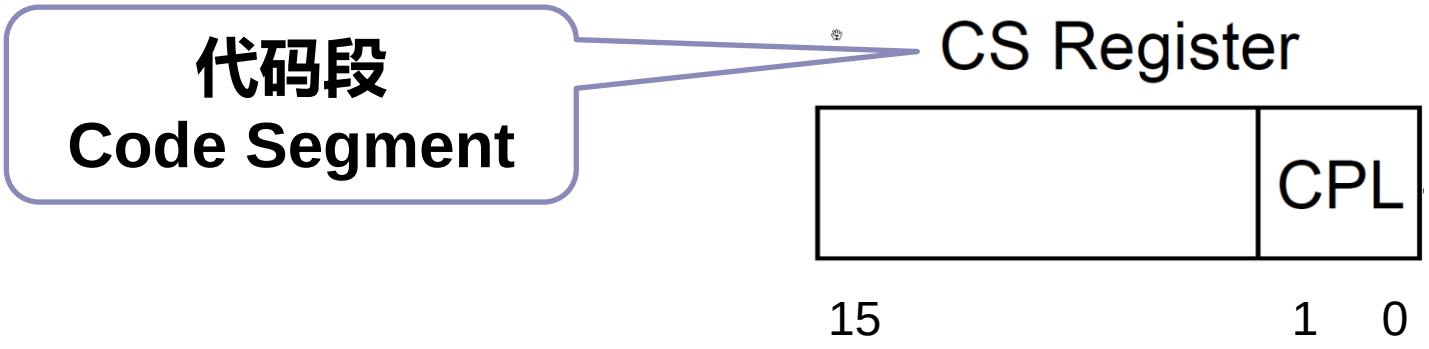


核心态和用户态

- 核心态是CPU的一种运行状态，在该状态下CPU可以执行**所有指令**（包括特权指令）
- 操作系统**内核**运行于核心态，用于程序运行于用户态

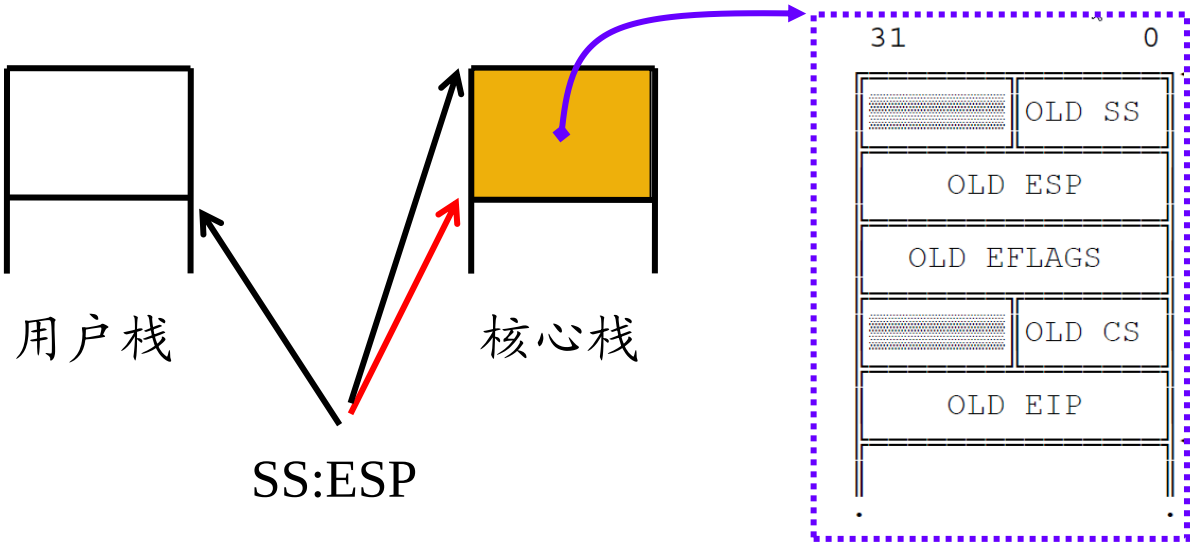
核心态和用户态

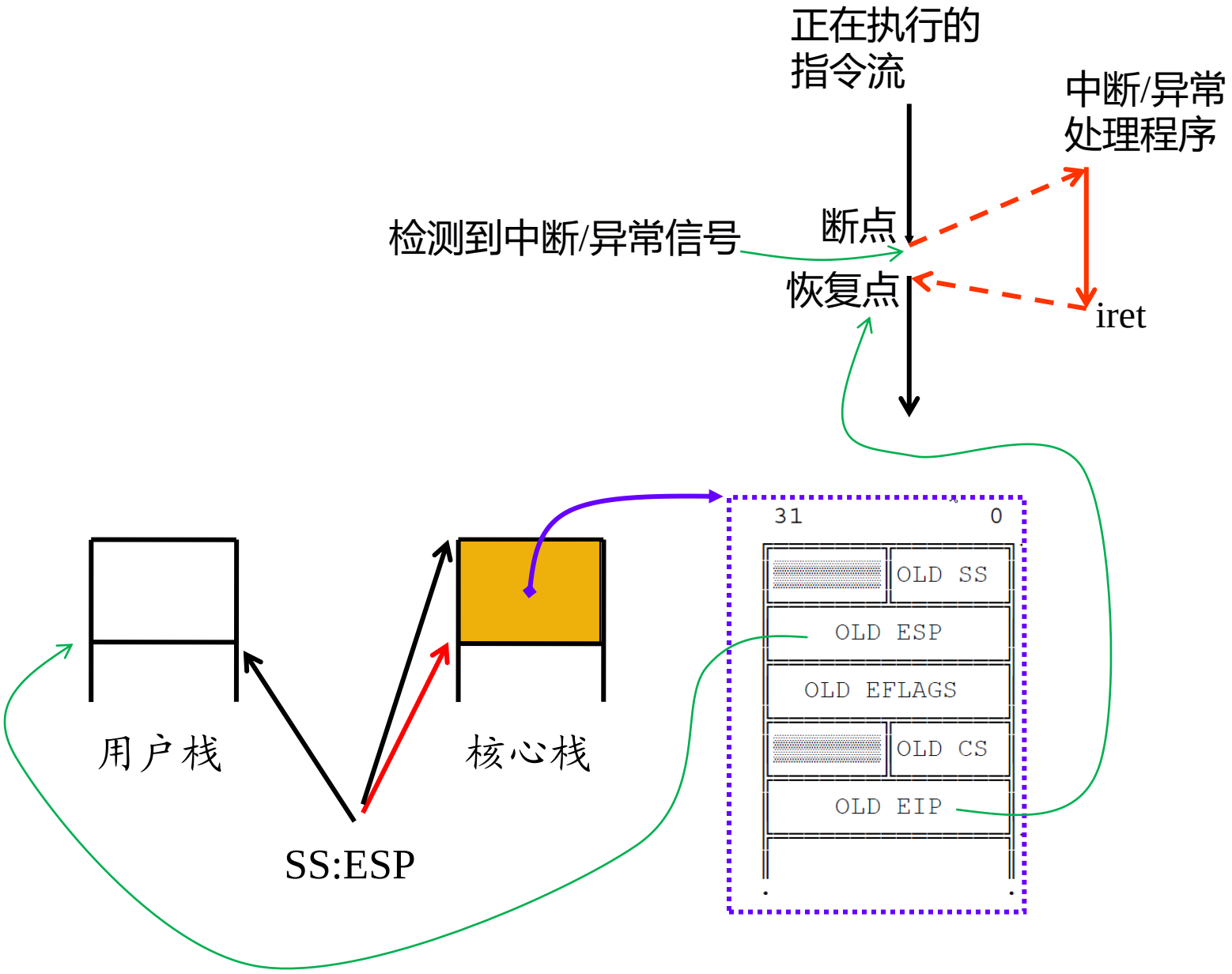
- 核心态是CPU的一种运行状态，在该状态下CPU可以执行所有指令（包括特权指令）
- **核心态的表示：CS寄存器的最低2位**
  - 00表示核心态；　11表示用户态



一. 中断/异常的处理过程

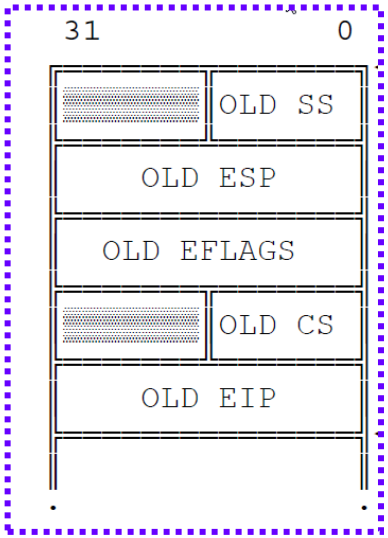
- 处理过程
  - CPU:
    a) 切换到核心栈，保存中断现场
    b) 查中断向量表，转到中断处理程序，同时切换到核心态




正在执行的
指令流
中断/异常
处理程序
检测到中断/异常信号
断点
恢复点
iret
31
0
OLD SS
OLD ESP
OLD EFLAGS
OLD CS
OLD EIP
用户栈
核心栈
SS:ESP

一. 中断/异常的处理过程

- 处理过程
  - CPU
    a) 切换到核心栈，保存中断现场
    b) 查中断向量表，转到中断处理程序，同时切换到核心态
  - 中断处理程序
    a) 中断处理
    b) iret (
    恢复中断现场，
    恢复用户栈和用户态)



目录

一. 中断/异常的处理过程

二. 演示：除零异常的响应

三. 独学&讨论

二. 演示

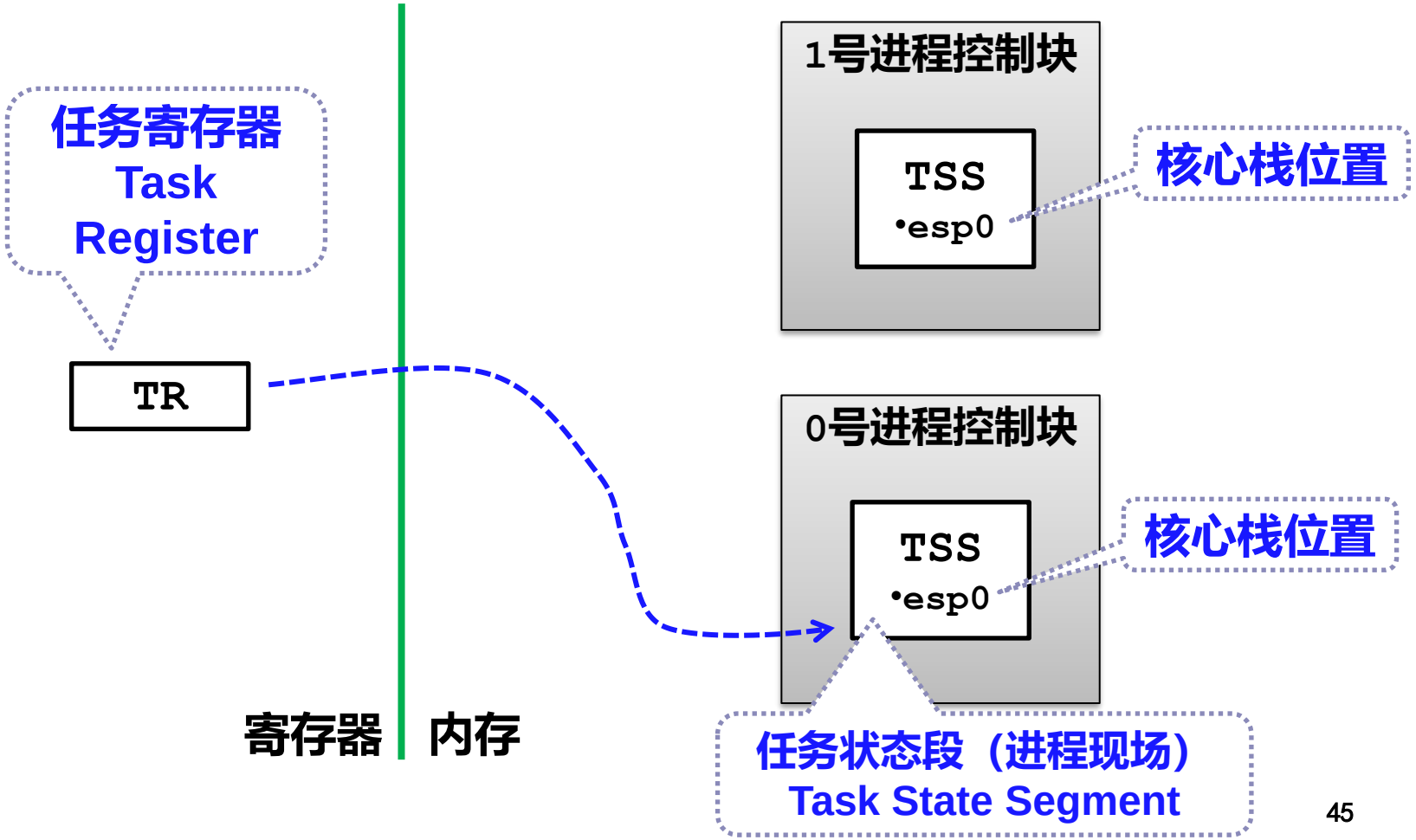
- 内容：

  除零异常的响应

  - 位置：idiv执行前后
  - 状态：cs:eip，ss:esp，核心栈

核心栈的位置

- 每个进程都有自己的核心栈



目录

一. 中断/异常的处理过程

二. 演示：除零异常的响应

三. 独学&讨论

独学&讨论

学习(386intel.pdf)：

- §9.4~9.6.1：P155始，IDT，IDT Descriptors，Interrupt Procedures
- LGDT/LIDT指令：P330始

问题：

- 不用单独的核心栈会有什么问题？
- 用户程序怎么才能利用中断/异常机制获得特权？

小结

一. 中断/异常的处理过程

栈的切换，态的切换，中断向量表，中断处理程序，iret

二. 演示：除零异常的响应

idiv指令，divide_error

三. 独学&讨论

Interrupt Procedures，LIDT指令

作业

- 实训：课堂练习2.2的第2~3关：int指令分析、iret指令分析