

中断和异常

日期: 2024 年 10 月 16 日

知识总览

- 中断的作用
 - 中断的类型
 - 内中断（也称“**异常**”）
 - 外中断（也称“**中断**”）
 - 中断机制的基本原理
-

中断的作用

- CPU 上会运行两种程序，一种是**操作系统的内核程序**，一种是**应用程序**
 - 操作系统的内核程序是整个系统的管理者
 - 在合适的情况下，操作系统内核会把 CPU 的使用权主动让给应用程序（第二章进程管理相关内容）
 - “中断”是**让操作系统夺回 CPU 使用权**的唯一途径
 - 如果没有“中断”机制，那么一旦应用在 CPU 上运行，CPU 就会一直运行这个应用程序，也就无法实现“并发”
 - **内核态 -> 用户态**：执行一条**特权指令**——**修改 PSW**的标志位为“用户态”，这个动作意味着操作系统将主动让出 CPU 使用权
 - **用户态 -> 内核态**：由“**中断**”引发，**硬件自动完成变态过程**，触发中断信号意味着操作系统将强行夺回 CPU 的使用权
-

中断的类型

- **内中断**：与当前执行的指令**有关**，中断信号来源于 CPU **内部**
- **外中断**：与当前执行的指令**无关**，中断信号来源于 CPU **外部**

内中断的例子

- **应用程序**
 - 指令 1
 - 指令 2
 - 指令 3（被黑客尝试植入的**特权指令**）
 - 指令 4

- **运行到指令 3 时**，CPU 检测到非法行为，立即中断信号转为内核态，操作系统夺回 CPU 使用权，并执行**异常处理程序**
- 试图在用户态下执行特权指令
- 执行除法指令时发现除数为 0
- 有时应用程序想请求操作系统内核的服务，此时会执行**一条特殊的指令——陷入指令**，该指令会引发一个内部中断信号
 - 执行“陷入指令”，意味着应用程序主动地将 CPU 控制权还给系统内核。“系统调用”就是通过陷入指令完成的

外中断的例子

- 外中断与当前执行的指令**无关**，中断信号来源于 CPU **外部**
- **时钟中断**——由时钟部件发来的中断信号
 - 时钟部件每隔一段时间片（如 50ms）会给 CPU 发送一个时钟中断信号
 - 中断后会切换到内核态，执行处理时钟中断的内核程序，操作系统内核决定接下来让另一个应用程序上 CPU 运行
 - 执行完内核程序后切换到被决定的第二个程序运行
 -如此循环
- **I/O 中断**——由输入/输出设备发来的中断信号
 - 当输入/输出任务完成时，向 CPU 发送中断信号，切换到内核态，执行相应的中断处理程序
- **每一条指令执行结束时**，CPU 都会例行检查是否有外中断信号

中断的分类

- **内中断**（也称异常、例外）：与当前执行的指令**有关**，中断信号来源于 CPU **内部**
 - **陷阱、陷入 (trap)**：由陷入指令引发，是程序故意引发的
 - **故障 (fault)**：由错误条件引起的，可能被内核程序修复。内核程序修复故障后会把 CPU 使用权还给应用程序，让它继续执行下去。如缺页故障 ---> Chapter 3
 - **终止 (abort)**：由致命错误引起，内核程序无法修复该错误，因此一般不再将 CPU 使用权还给引发终止的应用程序，而是直接终止该应用程序。如整数除 0、非法使用特权指令
- **外中断**（也称“中断”）：与当前执行的指令**无关**，中断信号来源于 CPU **外部**
 - 时钟中断
 - I/O 中断请求
- **狭义的中断仅指外中断**，“内中断”一般称为“异常”

中断机制的基本原理

- **不同的中断信号，需要不同的中断处理程序来处理。**当 CPU 检测到中断信号后，会根据中断信号的类型去查询“**中断向量表**”，以此来找到相应的中断处理程序在内存中的存放位置
 - 显然，中断处理程序一定是内核程序，需要运行在“内核态”
-

知识回顾与重要考点

• 中断和异常

◦ 中断的作用

- 让操作系统内核强行夺回 CPU 的控制权
- 使 CPU 从用户态变为内核态

◦ 中断的分类

- **内中断（也称异常、例外）**
 - 陷阱、陷入 (trap)
 - 故障 (fault)
 - 终止 (abort)
- **外中断（也称中断）**
 - 时钟中断
 - I/O 中断请求

◦ 中断机制的基本实现原理

- **检查中断信号**
 - **内中断：**CPU 在执行指令时会检查是否有异常发生
 - **外中断：**每个指令周期末尾，CPU 都会检查是否有外中断信号需要处理
- **找到相应的中断处理程序**
 - 通过“**中断向量表**”实现