中断/异常机制

- 中断/异常 对于操作系统的重要性
  就好比：汽车的发动机、飞机的引擎

→→ 可以说 操作系统
是由“中断驱动”或者“事件驱动”的

主要作用

- 及时处理设备发来的中断请求
- 可使OS捕获用户程序提出的服务请求
- 防止用户程序执行过程中的破坏性活动
- …… 等等

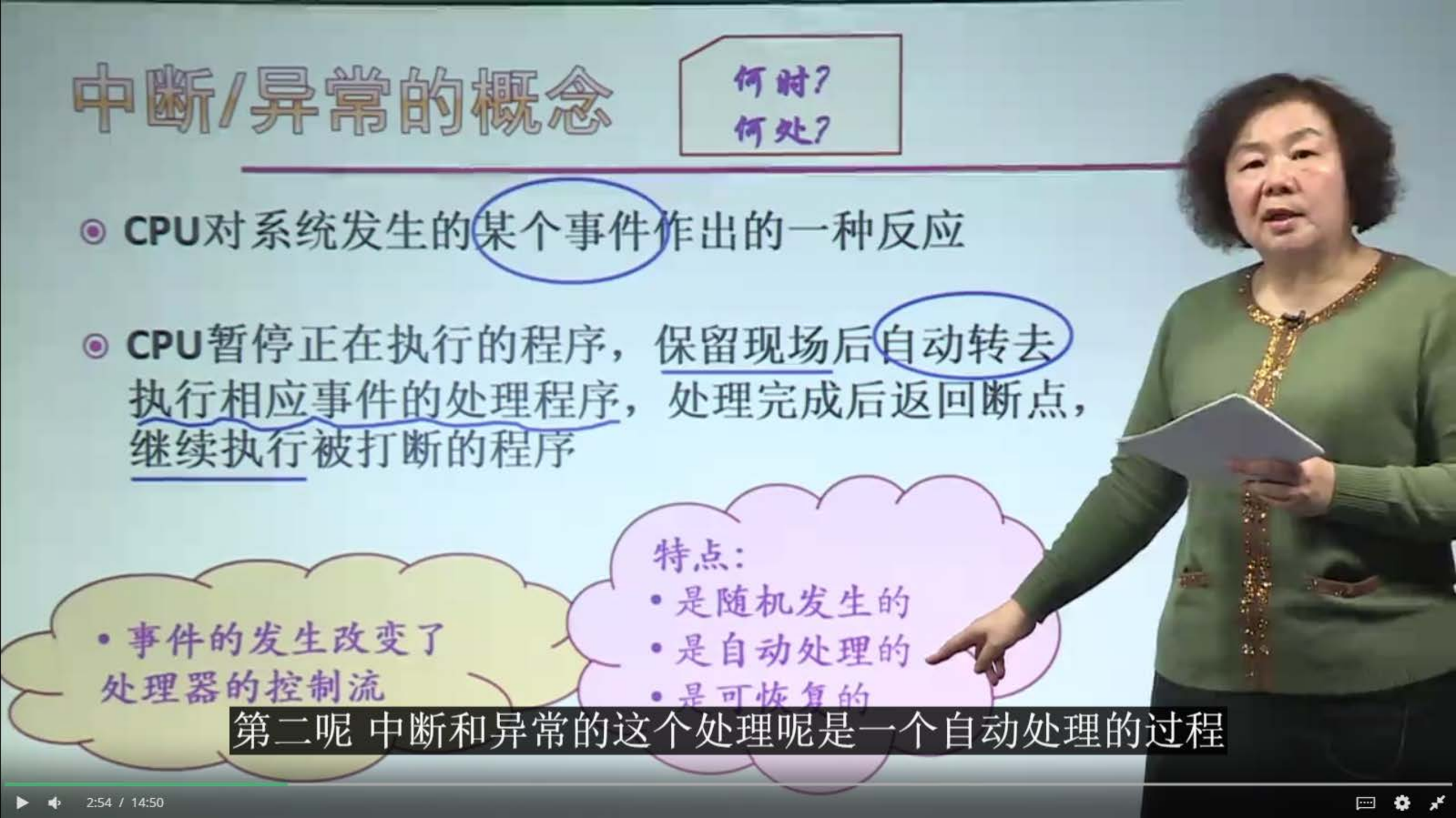
中断/异常的概念

何时?
何处?

- CPU对系统发生的某个事件作出的一种反应
- CPU暂停正在执行的程序，保留现场后自动转去执行相应事件的处理程序，处理完成后返回断点，继续执行被打断的程序

特点:
- 是随机发生的
- 是自动处理的
- 是可恢复的

- 事件的发生改变了处理器的控制流



第二呢 中断和异常的这个处理呢是一个自动处理的过程

中断/异常的概念

硬件完成
这一过程

- CPU对系统发生的某个事件作出的一种反应
- CPU暂停正在执行的程序，保留现场后自动转去执行相应事件的处理程序，处理完成后返回断点，继续执行被打断的程序

特点：
- 是随机发生的
- 是自动处理的
- 是可恢复的

- 事件的发生改变了处理器的控制流

硬件自动地完成整个控制流的转移工作 第三个特点呢是

中断/异常的概念

在以后某个时刻继续

- CPU对系统发生的某个事件作出的一种反应
- CPU暂停正在执行的程序，保留现场后自动转去执行相应事件的处理程序，处理完成后返回断点，继续执行被打断的程序

- 事件的发生改变了处理器的控制流

特点：
- 是随机发生的
- 是自动处理的
- 是可恢复的

为什么引入中断与异常？

历史背景

- 中断的引入：为了支持CPU和设备之间的并行操作
  - 当CPU启动设备进行输入/输出后，设备便可以独立工作，CPU转去处理与此次输入/输出不相关的事情；当设备完成输入/输出后，通过向CPU发中断报告此次输入/输出的结果，让CPU决定如何处理以后的事情

- 异常的引入：表示CPU执行指令时本身出现的问题
  - 如算术溢出、除零、取数时的奇偶错，访存地址时越界或执行了“陷入指令”等，这时硬件改变了CPU当前的执行流程，转到相应的错误处理程序或异常处理程序或执行系统调用

那么早期呢，是不区分 中断和异常的，都叫中断，我们其实都统称为中断


操作系统
要好好学习
今天要学习的
内容是中断
操作系统
北京大学
那么他作为一个 CPU 呢正在看书，那么去
今天的内容呢是学习中断，今天要学习的内容是中断
7:17 / 14:50


哎呀，来电话了！
接个电话先
再来看书
操作系统
北京大学
来电话了，电话铃响了，那么小明呢要先接个电话 然后呢再来看书。
7:32 / 14:50

那么这个时候他要 为了继续能够看书呢要在刚才被打断的地方呢要放个书签

当完成了这样接电话的任务处理之后呢

7:50 / 14:50

小明呢接着从刚才那个书签的地方接着来学习中断的知识 那么这就是中断。


操作系统
要好好学习
今天要学习的内容是异常
操作系统
北京大学
我们再来看看异常 小明今天要学呢是异常的这个概念
8:11 / 14:50

操作系统
要好好学习

今天要学习的内容是异常

口渴了
喝口水再来学

那么他决定先喝口水然后再来学

8:30 / 14:50


操作系统
要好好学习
今天要学习的内容是异常
在这里要放个书签
口渴了
喝口水再来学
操作系统
像刚才一样，他也要在 打断的地方放一个书签以便待会儿接着学习
8:37 / 14:50

那么喝水的过程，实际上就我们把它看成是一个异常处理的一个 执行过程。


异常
什么是异常？
异常就是...
接着学习
异常知识
操作系统
通过这么一个日常生活中的例子呢我们可能会体会到中断和异常的不同点
8:54 / 14:50

事件

中断：外部事件，正在运行的程序所不期望的

异常：由正在执行的指令引发

- 事件
  - 中断(外中断)
    - I/O中断
    - 时钟中断
    - 硬件故障
  - 异常(内中断)
    - 系统调用
    - 页故障/页错误
    - 保护性异常
    - 断点指令
    - (如算术溢出等)

举例

这样我们可以区别一下中断和异常 那么我们最后呢

中断与异常的小结

类别	原因	异步/同步	返回行为
中断 Interrupt	来自I/O设备、其他硬件部件	异步	总是返回到下一条指令
陷入Trap	有意识安排的	同步	返回到下一条指令
故障Fault	可恢复的错误	同步	返回到当前指令
终止Abort	不可恢复的错误	同步	不会返回

那么这张表呢把中断和异常做了一个很好的 小结

0:00 下面我们介绍中断与异常机制 那么这个机制是操作系统的一个 驱动力，非常重要。 操作系统当中 中断和异常机制就好比是汽车中的发动机，或者是飞机引擎 靠它来驱动操作系统的运行。 我们 有的时候是可以这么说的，操作系统是由中断驱动的 当然了也可以说是由事件驱动的。 那么有了这个机制 操作系统就可以做很多的事情，比如说 操作系统就可以及时地来处理各种各样的外部设备 发来的一些请求。 也可以使得操作系统 可以及时地捕获到了用户程序提出的各种服务 同时，有了这个机制还可以防止用户 在执行过程中有意识或者是无意识的一些破坏性的活动 等等，所以有了这样一个机制，操作系统就可以做更多的事情 我们来介绍一下中断与异常的概念 所谓中断与异常，实际上是 CPU 对系统发生的某个事件的一种反应 当这个事件发生的时候 通过对这个事件的处理 实际上是改变了一个控制流 我们通常说事件的发生改变了 CPU 的一个控制流 那么怎么改变呢？我们来看一下。 发生了事件之后 CPU 会暂停正在执行的程序 保留现场。 然后呢 自动去处理，去执行对这个事件的处理程序 完成了这个处理过程之后，返回到断点 继续执行刚才被打断的程序。 这是一个对 中断和异常这样一个事件发生之后 如何改变控制流的一个描述 从这个描述当中呢，其实我们会看到了 有几个重要的特点。 一个特点呢 是中断或异常这个事件呢是随机发生的 我们在何处发生？什么时候发生？ 事先呢是不能判断的。 第二呢 中断和异常的这个处理呢是一个自动处理的过程 所谓自动处理的过程呢我们是想强调这个过程是由硬件来完成的 硬件自动地完成整个控制流的转移工作 第三个特点呢是 中断和异常事件发生之后呢是可恢复的 所谓可恢复呢就指的是被打断的 这个事件完了之后被打断的这个程序可以在以后某个时刻呢再接着执行 所以这是对中断和异常的一个初步的介绍 那么我们通过历史的背景 来讲一下为什么引入中断与异常 首先，中断的引入 实际上是为了支持 CPU 与外部设备的一个并行操作这样一个能力 早期的计算机系统，如果没有中断机制的话 那么 CPU 要负责对设备的所有工作的管理 但是有了这样一个中断机制之后呢 我们就可以这样来工作了。 CPU 呢会去启动 输入输出设备的工作。 启动做完了之后呢 设备本身就可以独立自己进行工作了 而

CPU 这个时候呢可以转去处理一些和这次输入输出没有关系的事情 那么当然，当设备完成了它的任务之后 它通过什么样的方式告诉 CPU 呢？ 实际上是通过了中断。 那么外部设备 去通过中断向 CPU 报告我这次输入输出的结果 让 CPU 来决策下面该做什么事情。 所以它是一种 向 CPU 发中断呢是一种向 CPU 汇报的手段 这是最早为什么引入中断机制，是为了让 CPU 和设备能够并行工作 但是设备完成工作之后呢要向 CPU 汇报，所以引入了中断 那么异常呢？异常呢主要是 表示 CPU 在执行过程中自身出现的一些问题 比如说在执行过程中出现了算术溢出 除零，或者是取数的时候呢奇偶交元错 也可能访问内存的时候呢地址越界 或者是刻意安排了一个特殊的陷入指令 那么这个时候硬件呢会改变 CPU 的当前的一个控制流程 然后去转去执行对错误的处理 对异常的处理或者是执行相应的系统服务 那么早期呢，是不区分 中断和异常的，都叫中断，我们其实都统称为中断 但是随着它们发生的原因不同，处理的一些 过程不同，那么就有了中断和异常的这么一个区分 但是我们在讲课的过程中 在讨论的过程中，有的时候 在不影响它的，这个对它的理解的情况下 我们可能就用中断来代表了中断和异常的各种事件 有的时候我们会这样，这是为了简化一些说法 那么这是关于中断和异常为什么引入 下面让我们放松一下，我们举一个日常 生活中的例子来体会一下中断和异常的一个区别 小明选了这门课 那么他呢要好好学习操作系统 那么他作为一个 CPU 呢正在看书，那么去 今天的内容呢是学习中断，今天要学习的内容是中断 这个时候呢在学习的过程中，大家看到 来电话了，电话铃响了，那么小明呢要先接个电话 然后呢再来看书。 那么这个时候他要 为了继续能够看书呢要在刚才被打断的地方呢要放个书签 然后呢他就去接电话去了 那么接电话呢就相当于我们说的中断的处理过程 当完成了这样接电话的任务处理之后呢 小明呢接着从刚才那个书签的地方接着来学习中断的知识 那么这就是中断。 我们再来看看异常 小明今天要学呢是异常的这个概念 那么他非常刻苦地学习，花了很长时间 结果呢学习这个异常 在学习过程中呢突然觉得口渴了 那么他决定先喝口水然后再来学 像刚才一样，他也要在 打断的地方放一个书签以便待会儿接着学习 那么喝水的过程，实际

上就我们把它看成是一个异常处理的一个 执行过程。 当他喝完水之后，接着回去 继续来学习异常的知识 通过这么一个日常生活中的例子呢我们可能会体会到中断和异常的不同点 下面我们来看看对中断和异常的具体的 事件的划分。 中断和异常实际上呢我们都把它可以叫做事件 那么中断呢，也可以称为外中断 异常呢我们一般叫内中断，这也是可以这么说的 我们来看看有哪些中断的事件呢？ 有 I/O 中断，输入输出中断，有时钟中断 还有硬件故障中断。 希望大家对 不同的中断类型呢能举出一两个例子 我们来看一下 I/O 中断 比如说，我们在键盘上按了 Ctrl + C 比如说网络上来了一个包，网卡接收到一个新的包，数据包 比如说打印机结束了 或者是读盘结束了 那么这样一些事件的发生，其实呢都是带来的是 I/O 中断 那么时钟中断呢，比如说我们设定了一个定时器 到点了，上，CPU 上运行的程序它的时间片到了，那么这些呢我们都 称之为时钟中断。 硬件故障中断呢我们可以举一个例子，比如说笔记本电脑 快没电了，这个电池快没了，快用完了 消耗完了，那这时候它要报警，提示你赶紧存盘或者是 充电。 那么这呢就属于硬件故障中断 读内存的时候呢，奇偶校验错呢也是一个比较典型的硬件故障中断 下面我们来看看有哪些具体的异常的事件 异常呢有很多 我们刚才说的一个典型的异常，系统调用其实是属于异常的 还有页故障或者叫页错误异常 那么我们在第一讲当中介绍 HelloWorld 程序执行过程中 第一条指令执行的时候产生了一个缺页异常，因为 要执行的代码在磁盘上还没有读入内存，那么这个时候 执行到这里的时候会，硬件会产生一个 叫做页故障，那我们具体的页故障就是缺页异常 保护性异常。 比如说一块内存空间 它呢标记成只读。 当要做一个写的操作 这个就冲突了，操作和它的权限冲突了，所以这时候呢会引发一个保护性异常 如果要访问一个这个内存空间 那么访问的这个区域的地址越界了，那么也算保护性异常 还有就是断点指令，包括了我们说我们要调试程序，我们要设置 单步的调试，那么这样的话呢就要设置这样一个断点指令 你这样的话呢这就是一个异常。 当然还有很多的异常我们称之为程序性 异常。 程序性异常我们都比较常见，比如我们前面说的 算术溢出啊，除零啊，栈溢出等等都属于程序性异常 好，那么根据刚才那个日常中的例子和我们现在 举的例子我们可以归纳总结出来，所谓中断呢是外部事件 在 CPU 之外产生的事件打断了 CPU 那么这些事件呢是正在运行的程序所不期望的 异常呢是由正在执行的指令而引发的 这样我们可以区别一下中断和异常 那么我们最后呢 对中断和异常呢再做一个小结。 中断 实际上是

来自各种各样的外部设备和一些硬件部件 那么它是个异步的事件。在处理完中断之后 通常我们是返回到下一条指令 第一条指令执行完了，那么前一条指令执行完了 那么处理中断，返回到下一条指令执行 那么异常呢又分为三类 一类呢我们叫做陷入 一类叫做故障，一类叫做终止 那么陷入呢实际上是程序中刻意安排的，有意识安排的，比如说我们安排的那条陷入指令 这是一条特殊的向操作系统提出请求的指令，那么这是刻意安排的 那么在处理完相关的事件之后呢是返回到下一条指令 如果在执行这条指令的时候呢发生了故障 那么这个时候这个故障又是可恢复的，那么 去做处理，最后返回到刚才发生故障的这条指令，返回到当前指令 那么如果发生了 一个事件，一个故障，这个故障呢是 不可恢复的，那我们把它称之为终止，也就是说 当发生这样的事件，那么就不会返回了 就退出了，因为出错了，而且是不可恢复的错 那么这张表呢把中断和异常做了一个很好的 小结