

硬件做什么？ 软件怎么做？ .....

## 中断 / 异常机制工作原理

正如我们刚才所介绍的



# 中断/异常机制工作原理

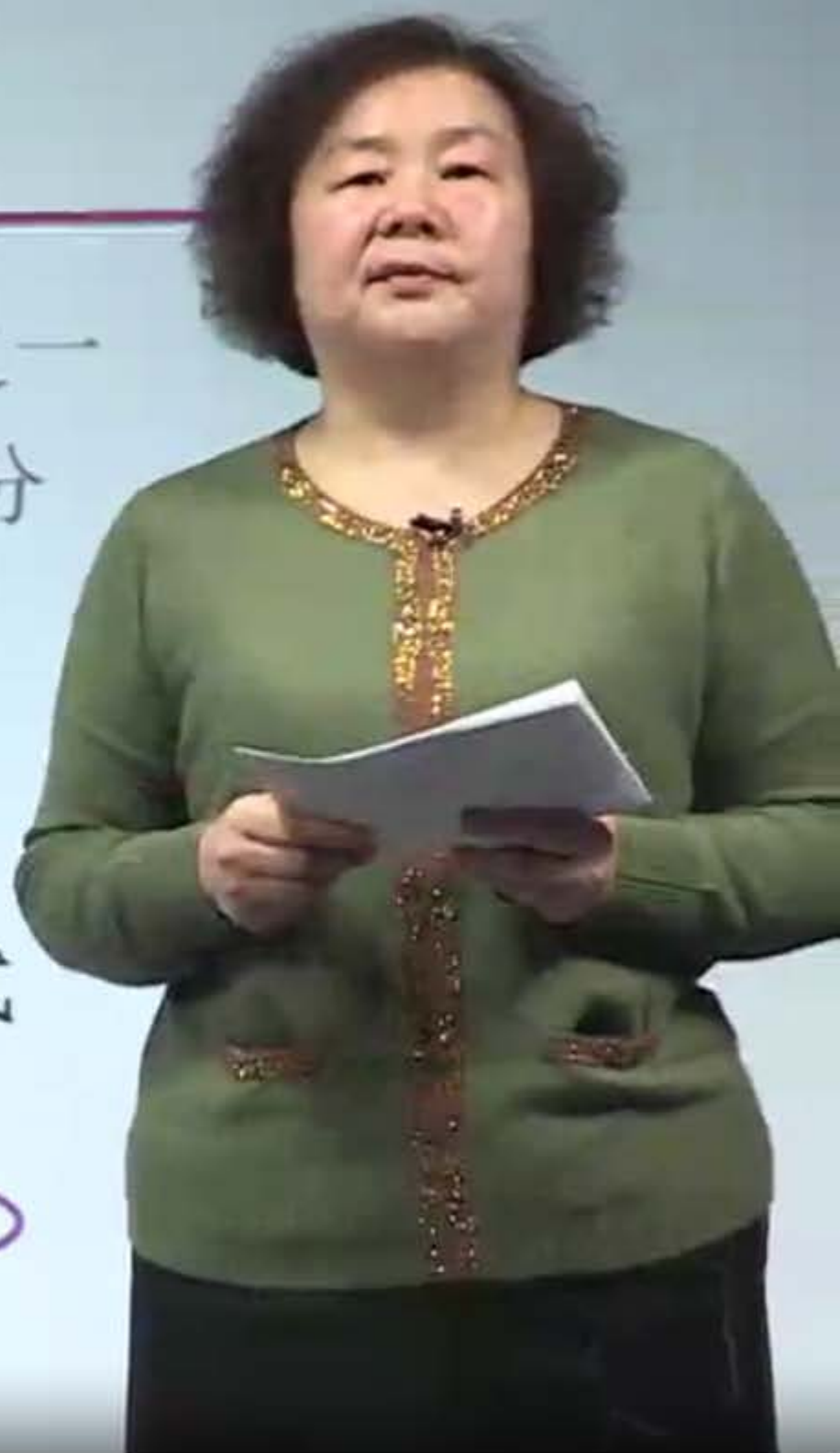
- 中断/异常机制是现代计算机系统的核心机制之一  
硬件和软件相互配合而使计算机系统得以充分发挥能力

- 硬件该做什么事？ —— 中断/异常响应

捕获中断源发出的中断/异常请求，以一定方式响应，将处理器控制权交给特定的处理程序

- 软件要做什么事？ —— 中断/异常处理程序

识别中断/异常类型并完成相应的处理





# 中断响应

中断响应:

发现中断、接收中断的过程  
由中断硬件部件完成

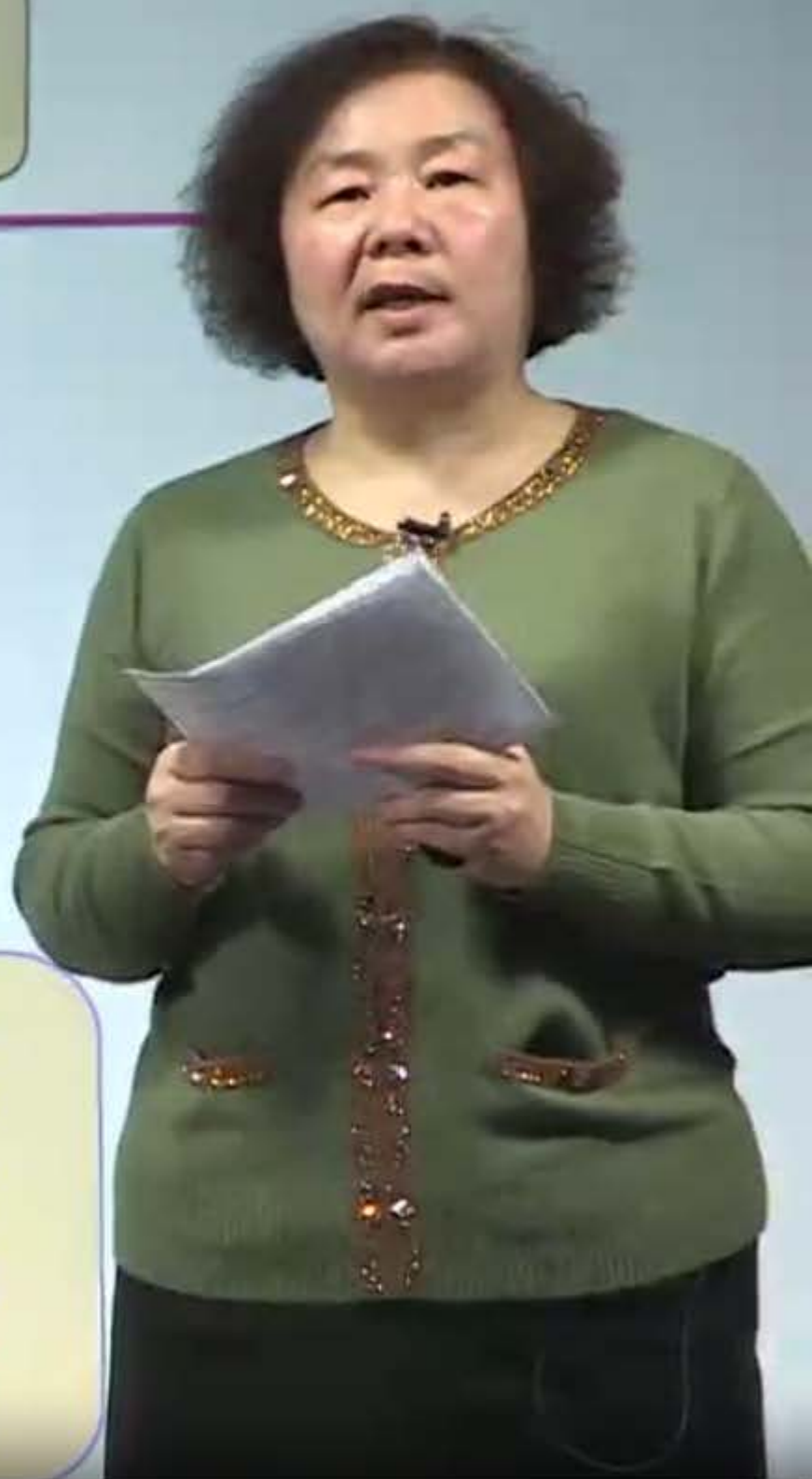
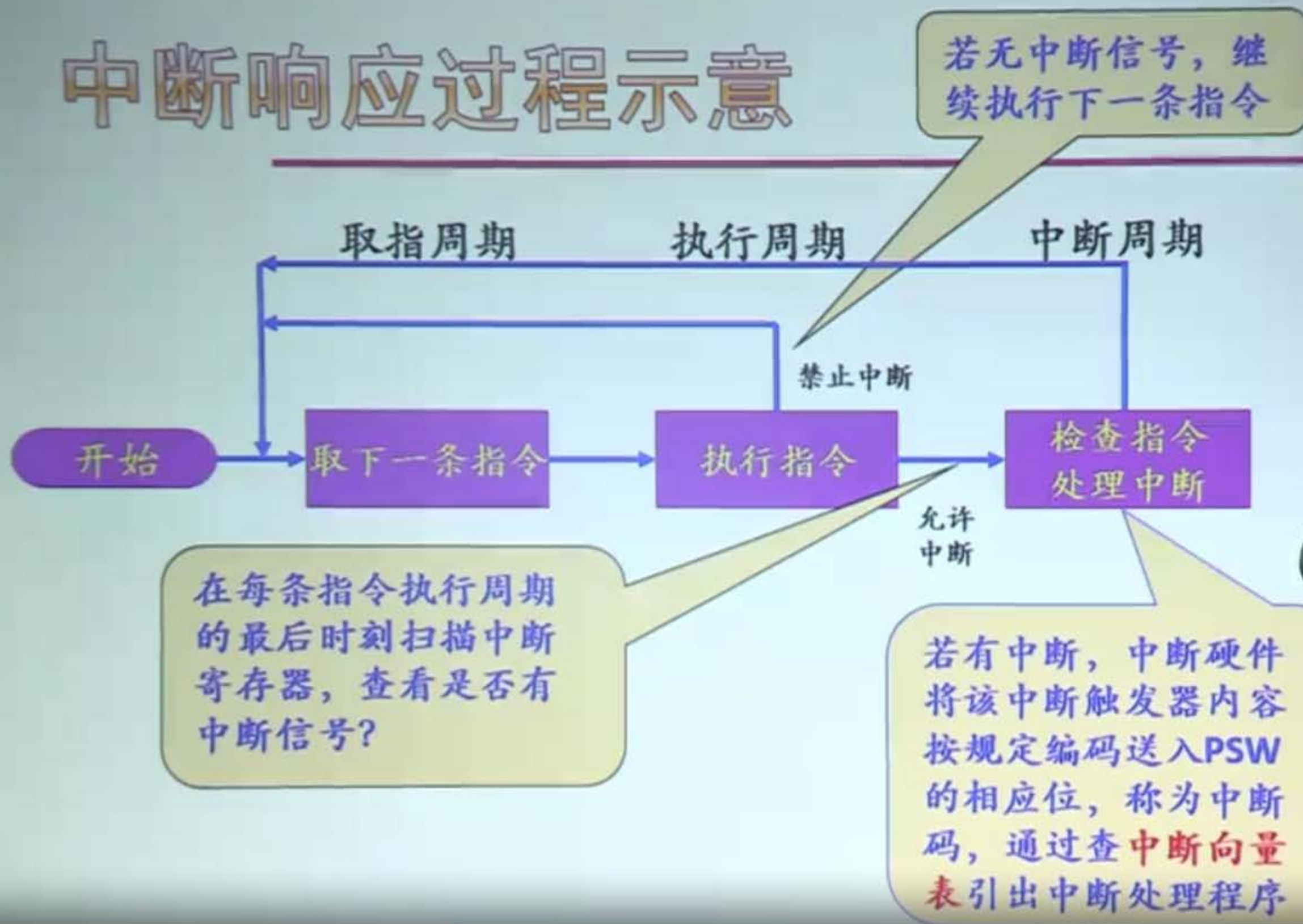
处理器控制部件中 设有 中断寄存器

CPU何时响应中断?

CPU 何时来做响应中断的工作呢?



# 中断响应过程示意

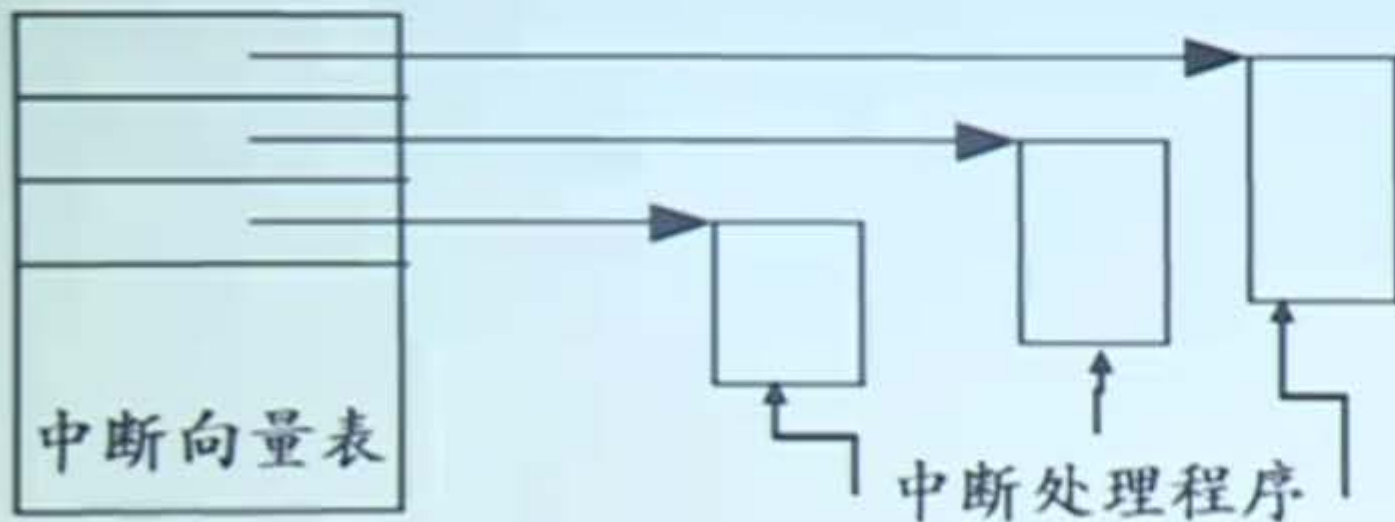




# 中断向量表

## ◎ 中断向量:

一个内存单元，存放中断处理程序入口地址和程序运行时所需的处理机状态字



执行流程按中断号/异常类型的不同，通过中断向量表转移控制权给中断处理程序



# LINUX中的中断向量表

向量范围	用途
0~19	不可屏蔽中断和异常
20~31	Intel保留
32~127	外部中断 (IRQ)
128 (0x80)	用于系统调用的可编程异常
129~238	外部中断
239	本地APIC时钟中断
240	本地APIC高温中断
241~250	Linux保留
251~253	处理器间中断
254	本地APIC错误中断
255	本地APIC伪中断

不可屏蔽中断/异常

- 0 - 除零
- 1 - 单步调试
- 4 - 算术溢出
- 6 - 非法操作数
- 12 - 栈异常
- 13 - 保护性错误
- 14 - 缺页异常





# 中断响应示意图

① 设备发中断信号

③ 根据中断码查表

⑤ 执行中断处理程序

② 硬件保存现场

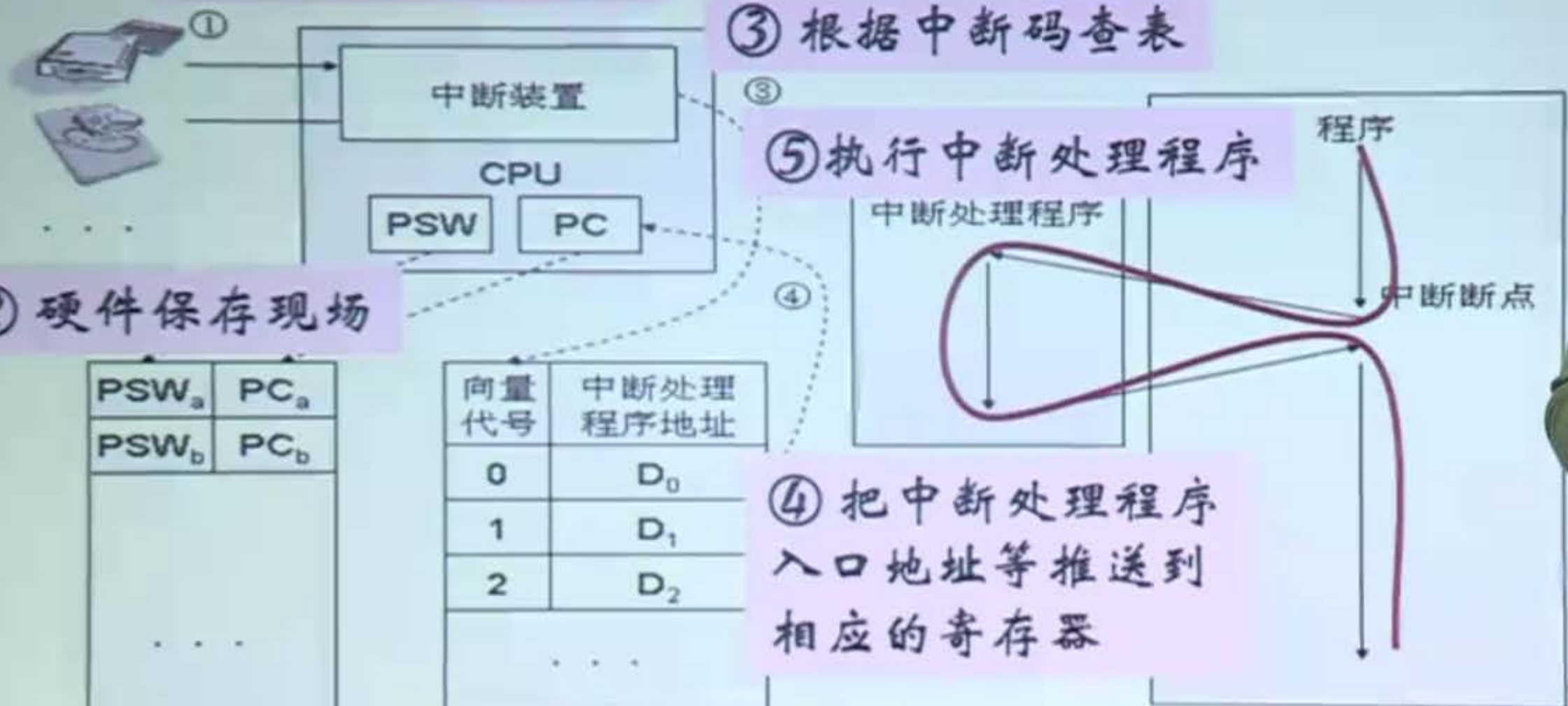
PSW <sub>a</sub>	PC <sub>a</sub>
PSW <sub>b</sub>	PC <sub>b</sub>
...	

系统堆栈

向量代号	中断处理程序地址
0	D <sub>0</sub>
1	D <sub>1</sub>
2	D <sub>2</sub>
...	
255	D <sub>255</sub>

中断向量表

④ 把中断处理程序入口地址等推送到相应的寄存器



# 中断处理程序

- ◎ 设计操作系统时，为每一类中断/异常事件编好相应的处理程序，并设置好中断向量表
- ◎ 系统运行时若响应中断，中断硬件部件将**CPU**控制权转给中断处理程序：
  - 保存相关寄存器信息
  - 分析中断/异常的具体原因
  - 执行对应的处理功能
  - 恢复现场，返回被事件打断的程序





# 中断处理程序

- 设计操作系统时，为每一类中断/异常事件编好相应的处理程序，并设置好

软件提前做好  
硬件部件来执行

- 系统运行时若响应中断，制权转给中断处理程序：
  - 保存相关寄存器信息
  - 分析中断/异常的具体原因
  - 执行对应的处理功能
  - 恢复现场，

软件提前做好的，硬件部件来执行





# 中断/异常机制小结 (1/2)

以设备输入输出中断为例:

- 打印机给**CPU**发中断信号
- **CPU**处理完当前指令后检测到中断, 判断出中断来源并向相关设备发确认信号
- **CPU**开始为软件处理中断做准备:
  - 处理器状态被切换到内核态
  - 在系统栈中保存被中断程序的重要上下文环境, 主要是**程序计数器PC**、**程序状态字PSW**



硬件



硬件





# 小结 (2/2)

硬件

- ◎ CPU根据中断码查中断向量表，获得与该中断相关的处理程序的入口地址，并将PC设置成该地址，新的指令周期开始时，CPU控制转移到中断处理程序
- ◎ 中断处理程序开始工作
  - 在系统栈中保存现场信息
  - 检查I/O设备的状态信息，操纵I/O设备或者在设备和内存之间传送数据等等
- ◎ 中断处理结束时，CPU检测到中断返回指令，从系统栈中恢复被中断程序的上下文环境，CPU状态恢复成原来的状态，PSW和PC恢复成中断前的值，CPU开始一个新的指令周期

软件

硬件





# 举例：I/O中断处理程序

通常分为两类处理：

## ◎ I/O操作正常结束

- 若有程序正等待此次I/O的结果，则应将其唤醒
- 若要继续I/O操作，需要准备好数据重新启动I/O

## ◎ I/O操作出现错误

- 需要重新执行失败的I/O操作
- 重试次数有上限，达到时系统将判定硬件故障





下面我们介绍一下中断/异常机制的工作原理 主要是从硬件做什么？软件怎么做？来区分 正如我们刚才所介绍的 中断/异常机制，实际上是现代计算机系统中的核心机制之一 那么它的主要工作，是 硬件和软件相互配合来完成的 通过软硬件的配合，来使计算机的能力 得到充分地发挥 那么在这个机制工作过程中，硬件该做什么事情呢？硬件做的工作呢，实际上是捕获 中断源发出的各种中断/异常的请求 以某一方式呢来响应，通过 把控制权转交给特定的处理程序来完成这个过程 那么这个过程呢，我们把它归结为中断 异常的响应，响应中断，响应异常 这是硬件完成的工作。软件要做什么事情呢？软件做的事情就是，识别中断啊、异常类型 完成对应的处理，实际上就是 处理程序，我们把它称之为中断/异常处理程序。所以硬件完成的是 响应，软件完成的是处理 那么关于中断响应 它的主要的工作，就是发现中断 接收中断的一个过程，那么这个过程是由中断的硬件部件完成的 在我们的处理器控制部件当中呢 设置了一个中断寄存器 保存了发来的各种各样的中断信号 那么 CPU 何时来做响应中断的工作呢？我们来通过一个图来 示意，看看 CPU 是如何来响应中断的 CPU 取下一条指令 然后呢执行这条指令 在执行完这条指令之后，也就是，在每条指令周期 的最后，扫描中断寄存器 查看是否有中断发生 是否有中断信号 如果有中断信号，就要对中断进行相应的处理 所以，如果有中断 那么，中断的硬件，会将该中断触发器 它的内容按规定编码 送到了程序状态字寄存器当中的相应的位 那么这个中断，这几个规定的这个 码子，我们称之为中断码，中断码 所以把中断的信号按照规定编码，形成一个中断码，送到相应的 寄存器的相应位，然后硬件会去查一个表，中断向量表 查中断向量表呢是把中断处理程序把它调出来 啊，调出来。那么调出来以后 那么下一个周期，实际上就是执行中断处理程序 那么如果没有中断来 当然也可能说我不接收中断，那么就继续执行下一条指令 那么这里头我们看到硬件还要做一件事情，就是查一张表 中断向量表，那么这个中断向量表呢 实际上是个非常重要的软硬件结合的这么一个数据结构 我们来介绍一下什么是中断向量表 中断向量表，每一行呢实际上是一个中断向量 中断向量表是由若干中断向量组成 每一个中断向量其实就是一个内存单元 它是存放了中断处理程序的这个入口地址 以及这个程序，在运行的时候所需要的一个处理机的状态字 若干中断向量构成了中断向量表 那么，操作系统呢要把这张表填写 事先先编好了中断处理程序 把这些中断处理程序的起始地址，或者入口地址，填在中断向量表里头 那么当中断发

生的时候呢，那么执行流程就会按照中断的号儿，或者是异常的类型，不同，通过这张表，通过中断向量表把控制权转移给对应的中断处理程序。我们来用一个例子看一看 LINUX 当中，中断向量表我们得到一些感性认识。那么这一张中断向量表里有 0 到 255 个中断向量。其中呢，0 到 19 呢，是用于不可屏蔽中断和异常的。我们再进一步看看 0 到 19 当中，有哪些我们熟悉的 中断或异常，比如说 0 号，中断向量，其实就是除零异常，1 号中断向量呢是单步调试用的 4 号呢，是用于算术溢出 6 号呢，用于我们所说的一些非法操作数 12 号用于栈的异常 栈出现问题了。13 号用于保护性错误，保护性异常而 14 号呢，就用于我们说的缺页异常，页故障。那这些呢，我们都是之前，相对来讲都了解这样一些异常了。我们再看一个我们非常关注的，就是 128 号中断向量，128 号中断向量用于什么呢？大家可以看，用于我们的系统调用。用于，它也是一个异常，它是用于系统调用这样一个可编程的异常。所以通过这张表我们大概对中断向量表的构成有一个感性认识。下面我们通过这张图，来回顾一下整个中断响应的过程。第一步，设备发来了中断信号，第二步 中断硬件部件，会去保存现场。把这些内容保存到了系统堆栈里头，保存的内容我们也可以看到。主要的是 PSW 加 PC 的内容。然后，第三步，中断硬件根据中断码来去查中断向量表。通过查中断向量表得到对应的中断处理程序。第四步 把中断处理程序的入口地址等相关信息推送到相应的寄存器，那么下一个指令周期开始，就是执行中断处理程序了。我们从这张图可以看到一个程序正在执行过程中被打断了，被打断那一点叫做中断的断点。然后转去执行相应的中断处理程序。执行完中断处理程序之后呢，返回到刚才的断点，继续执行。所以整个中断的响应的过程，实际上就是这样一个示意。下面我们来看看软件部分。中断处理程序，在设计操作系统的时候，我们会为每一类中断或者异常事先都编好相应的处理程序。然后呢，在操作系统初始化的过程中呢，再把它这个中断向量表设置好，就把这些处理程序的入口地址和 PSW 这些信息设置到中断向量表里头。好，当软件设置好了之后，操作系统初始化以后，也建好了这张表。在运行的过程中，当系统运行时。要响应中断了，那么中断的硬件部件通过刚才我们描述



的步骤，就把 CPU 的控制权，转交给了中断控制程序 那么下面，就是中断处理程序主要做的工作 中断处理程序呢，首先要保存相关寄存器的信息 那么我们要说明一下，在刚才那张图当中 硬件部件会保存关键寄存器的信息 那么其他的一些寄存器的信息呢，还要再做进一步地保存 所以呢，中断处理程序呢，首先要做这样一个事儿 然后就去分析中断的异常的具体原因，究竟是哪一类中断，哪一类异常 比如说，打印机为例，打印机来了中断信号 那么具体的原因是打印机正常地完成了任务？还是 由于没有打印纸了，或者是没有碳粉了，来发出的警告 好，所以这是具体原因。根据这个具体原因呢，实际上呢，要执行对应的处理程序。啊，执行对应的处理程序。当然如果有些中断发生了严重的错误可能就不再返回了，就报错退出了 大部分情况下呢，执行完处理程序之后呢，会恢复现场 返回被事件打断的这个程序。归纳一下呢 实际上呢我们可以这样来总结 软件提前设置好，硬件部件来执行 所以我们要区分，哪些是软件，就是操作系统事先做好的工作 哪些是硬件，在执行过程中，要做的规定的一些流程 我们为了加深大家的印象，我们再次以输入输出中断为例 把这个过程再贯穿一次。比如说 打印机，给CPU发了中断信号 那么CPU处理完当前指令之后呢，就去检测到了这个中断 判断出中断的来源，并向相关的设备发了一个确认信号 那么这部分工作呢，肯定是由硬件来完成的 CPU开始为软件处理中断呢，来做准备工作 比如说要将CPU的状态切换到内核态 并且呢，在系统堆栈中呢保存非常重要的 寄存器内容，这里头主要指的是程序计数器PC，呃，PS，呃，程序计数器PC 还有这个程序状态字PSW 那么这部分工作呢也是由硬件做的 下面CPU根据中断码，来查询 中断向量表，获得与这个中断相关的处理程序的入口地址 并且将程序计数器PC设置成该地址 这样新的指令周期开始的时候，CPU的控制权，就交给了这个中断处理程序 那么，这呢，也是由硬件来完成的 做完这个事情之后，那么下一个中断处理程序开始工作了，那么这是由软件来完成的 那么当然了，包括在系统堆栈中保存一些现场信息 或者，如果因为我们是打印机中断，对吧，打印机完成了，那么我们来检查一下I/O设备的状态 操纵I/O设备，或者是在设备和内存之间呢来传送数据 啊，做这样一些比较具体的工作。当中断处理 结束了以后，那么CPU去执行的是中断返回指令 那么从系统栈中恢复 被中断程序的上下文环境，就是几次压栈的那种，把它恢复 那么CPU的状态呢，也是恢复成原来的状态，也要如果是从用户态进到核心态呢 那么就把它设置成用户态 而PSW和PC的值呢，也恢复成以

前断点的值 在一个新的指令周期开始呢，就继续执行被打断的那个程序 那这一部分工作呢，实际上也可以说是由硬件来完成的 所以在这一部分里头呢，我们要搞清楚 在中断响应，中断处理的过程当中 哪些是硬件做的，哪些是软件做的 那么软件呢 做的什么事情呢，我们可以用一个例子，啊，举个例子，来说明一下 中断处理程序都做哪些工作？比如说还是以 打印机为例，这是一个典型的I/O的中断处理程序 那么，通常呢，我们会做两类处理 一类呢，是I/O操作正常结束 正常结束呢，那你要做的事情就是说，如果有程序正在等待这次I/O的结果 那么就把它唤醒，让它继续 做要做的工作。如果有新的I/O操作在等待，要继续新的 I/O操作，那么你就准备好数据，重新启动I/O 就是这样一个正常结束。如果出现了错误，在I/O过程中出现了错误 那么，首先需要的是重复执行一下刚才失败的这种I/O操作 看看是不是偶然发生了错误，所以是要重新执行 但是这个重新执行呢是有上限的 比如说我重新执行两次或者三次 达到了上限之后呢，系统呢就是报错了，啊报错说出现故障了 那么通常呢，我们在比如读盘的时候，比如特别是 典型的我们看读光盘的时候，啊，读这个DVD的时候 如果DVD没读出来，那么会反复地读很多次，我们会感觉到时间很长，这样的话呢说明什么呢，它正在重复在执行 这就是软件一个典型的这个软件做的工作。