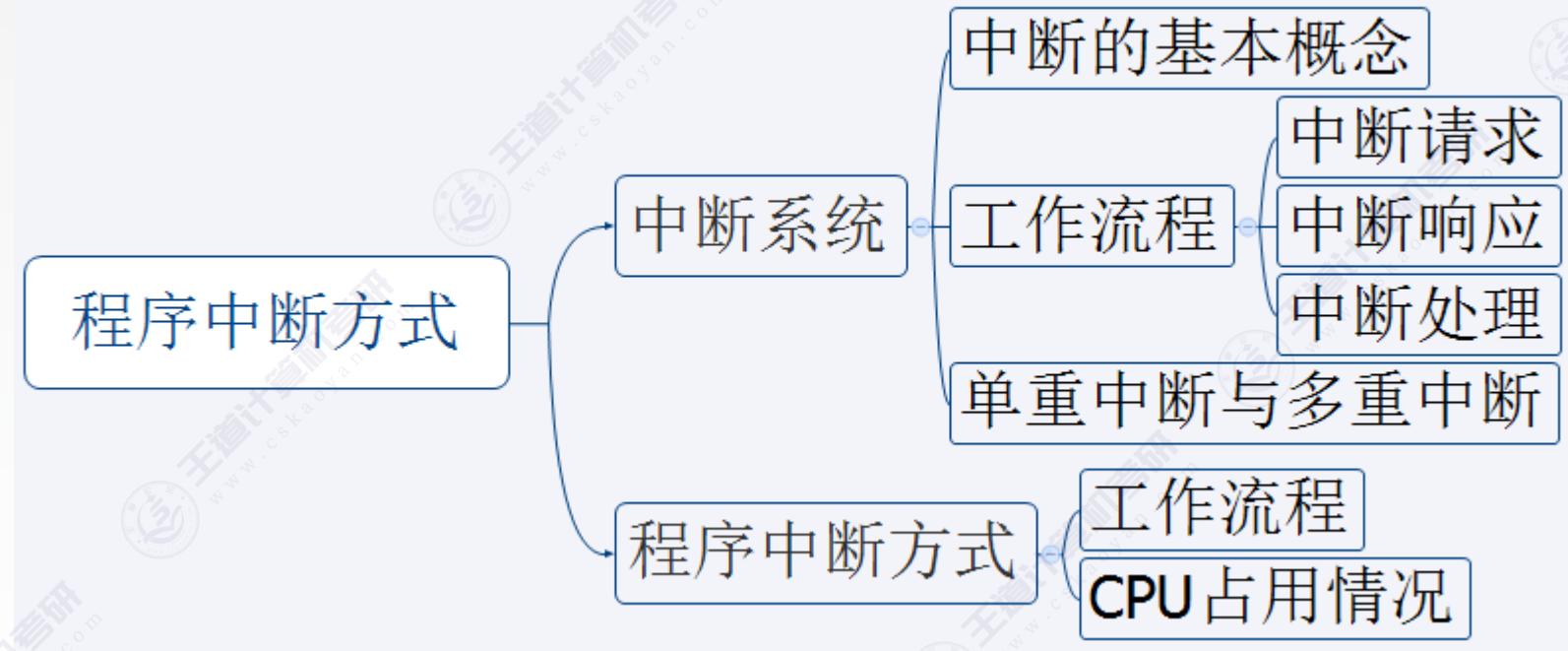


本节内容

输入/输出 系统

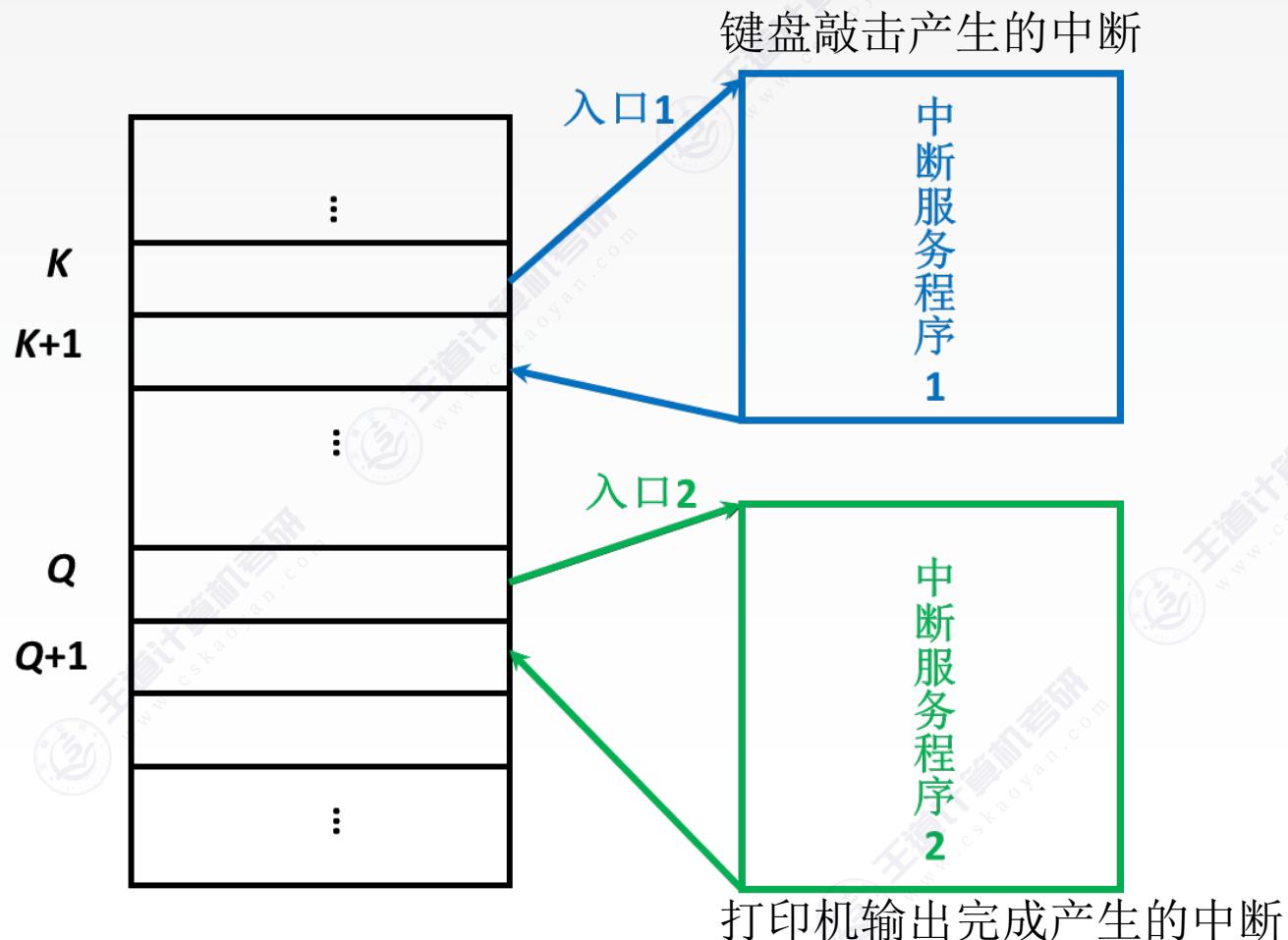
I/O方式2 程序中断方式

本节总览

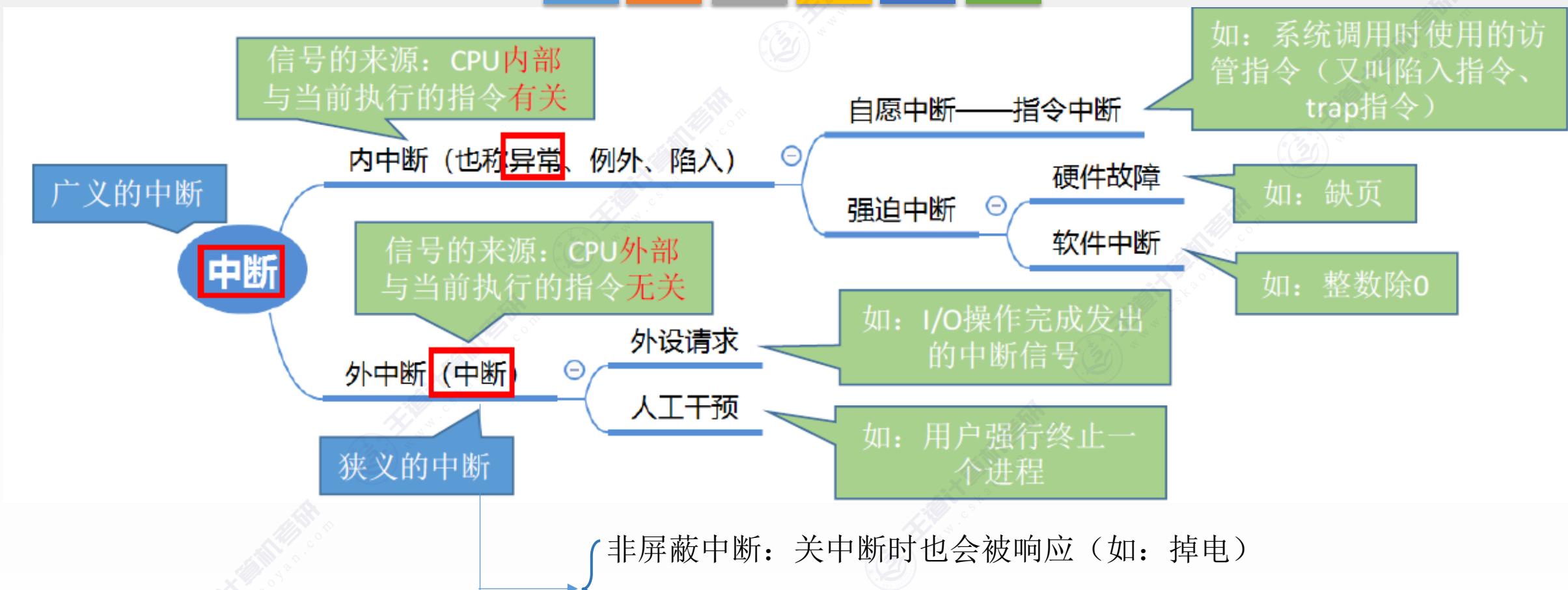


中断的基本概念

程序中断是指在计算机执行现行程序的过程中，出现某些急需处理的异常情况或特殊请求，CPU暂时中止现行程序，而转去对这些异常情况或特殊请求进行处理，在处理完毕后CPU又自动返回到现行程序的断点处，继续执行原程序。



中断请求的分类



关中断的作用: 实现原子操作

IF=1表示开中断 (允许中断)

IF=0表示关中断 (不允许中断)

IF: Interrupt Flag, 存在PSW中, 8086芯片的PSW如下

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

中断请求标记

如何判断是哪个设备
发来的中断信号？

每个中断源向CPU发出中断请求的时间是随机的。

为了记录中断事件并区分不同的中断源，中断系统需对每个中断源设置中断请求标记触发器INTR，当其状态为“1”时，表示中断源有请求。

这些触发器可组成中断请求标记寄存器，该寄存器可集中在CPU中，也可分散在各个中断源中。

中断请求标记寄存器	INTR ₁	INTR ₂	INTR ₃	INTR ₄	...	INTR _{n-1}	INTR _n
	0	0	0	1		0	1
	掉电	过热	鼠标	键盘		扫描仪	打印机

对于外中断，CPU是在统一的时刻即每条指令执行阶段结束前向接口发出中断查询信号，以获取I/O的中断请求，也就是说，CPU响应中断的时间是在每条指令执行阶段的结束时刻。

CPU响应中断必须满足以下3个条件：

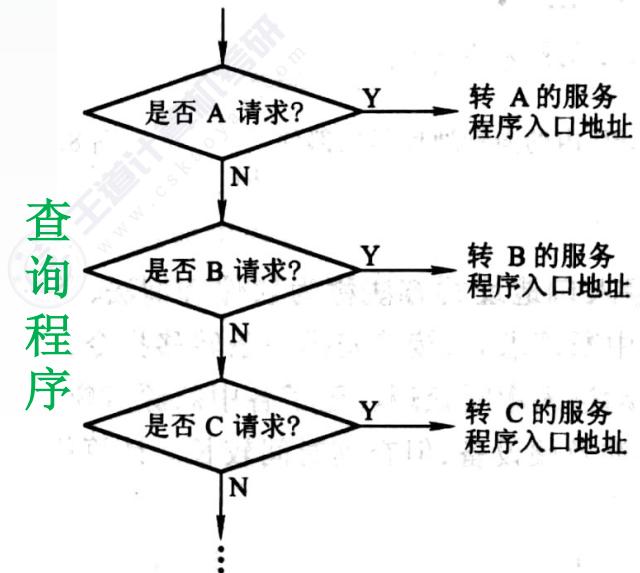
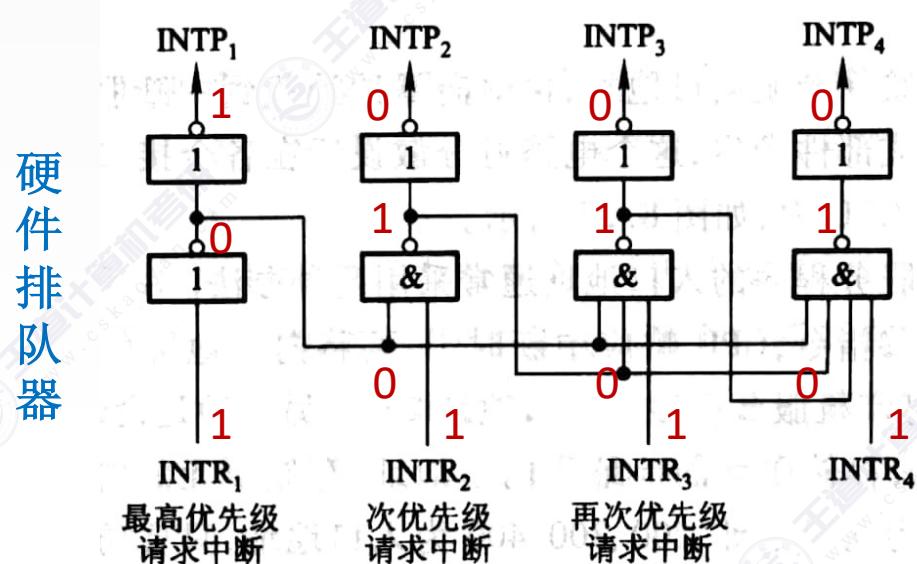
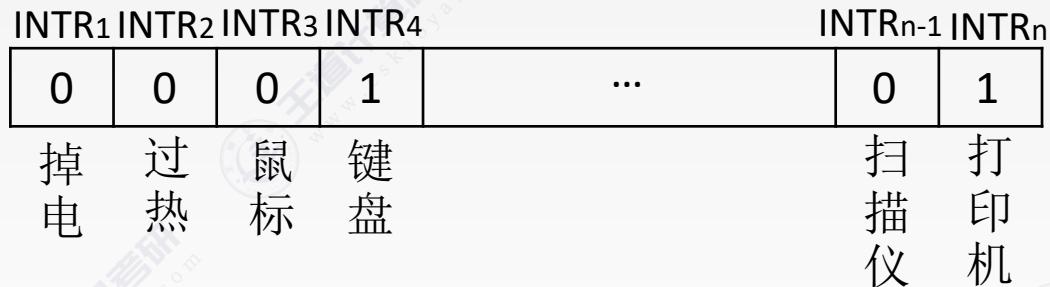
- ① 中断源有中断请求。
- ② CPU允许中断即开中断。
- ③ 一条指令执行完毕，且没有更紧迫的任务。

中断判优-实现

有多个中断信号同时到来，先处理哪个？

中断判优既可以用硬件实现，也可用软件实现：

硬件实现是通过**硬件排队器**实现的，它既可以设置在CPU中，也可以分散在各个中断源中；
软件实现是通过**查询程序**实现的。

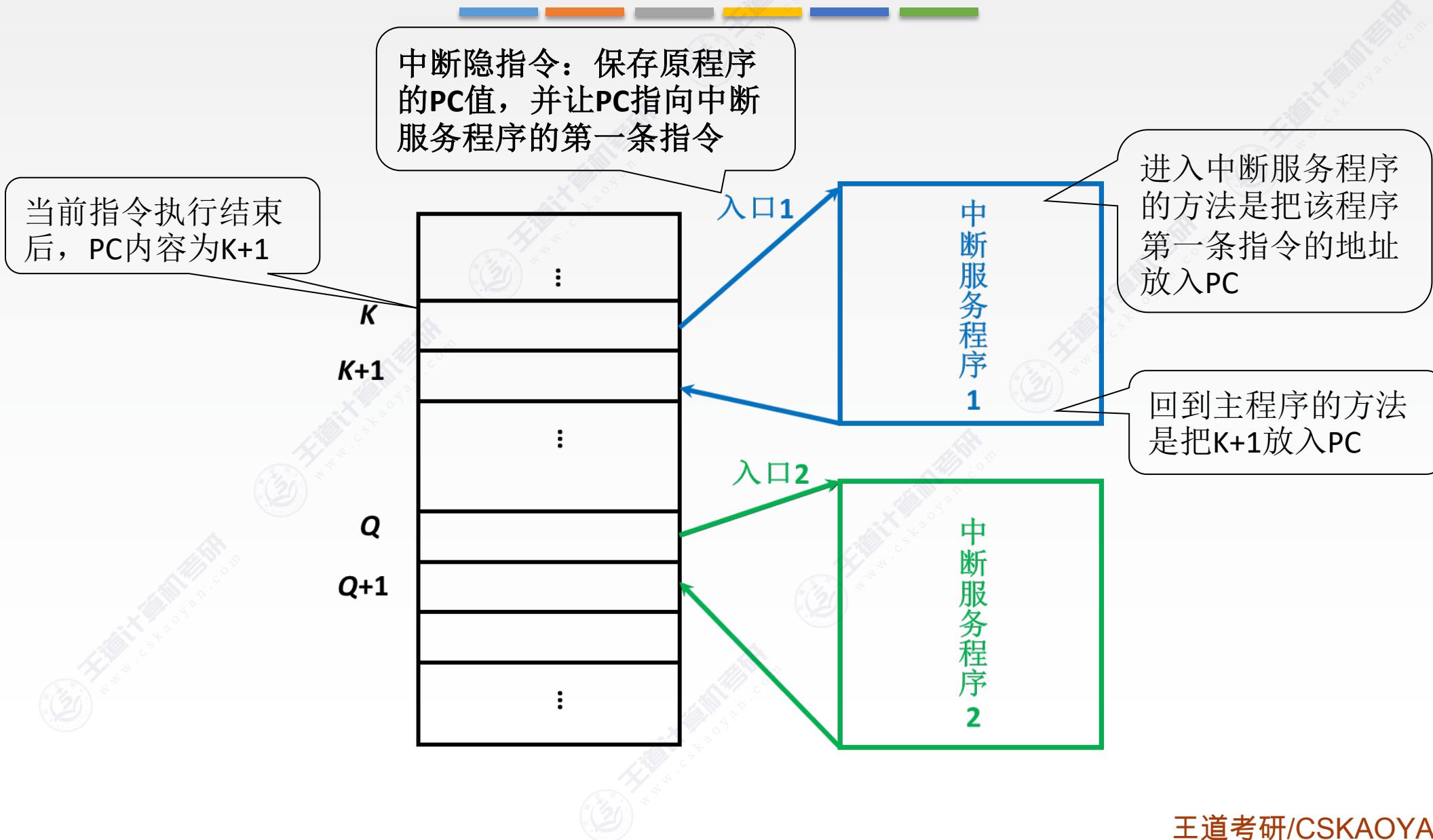


中断判优-优先级设置



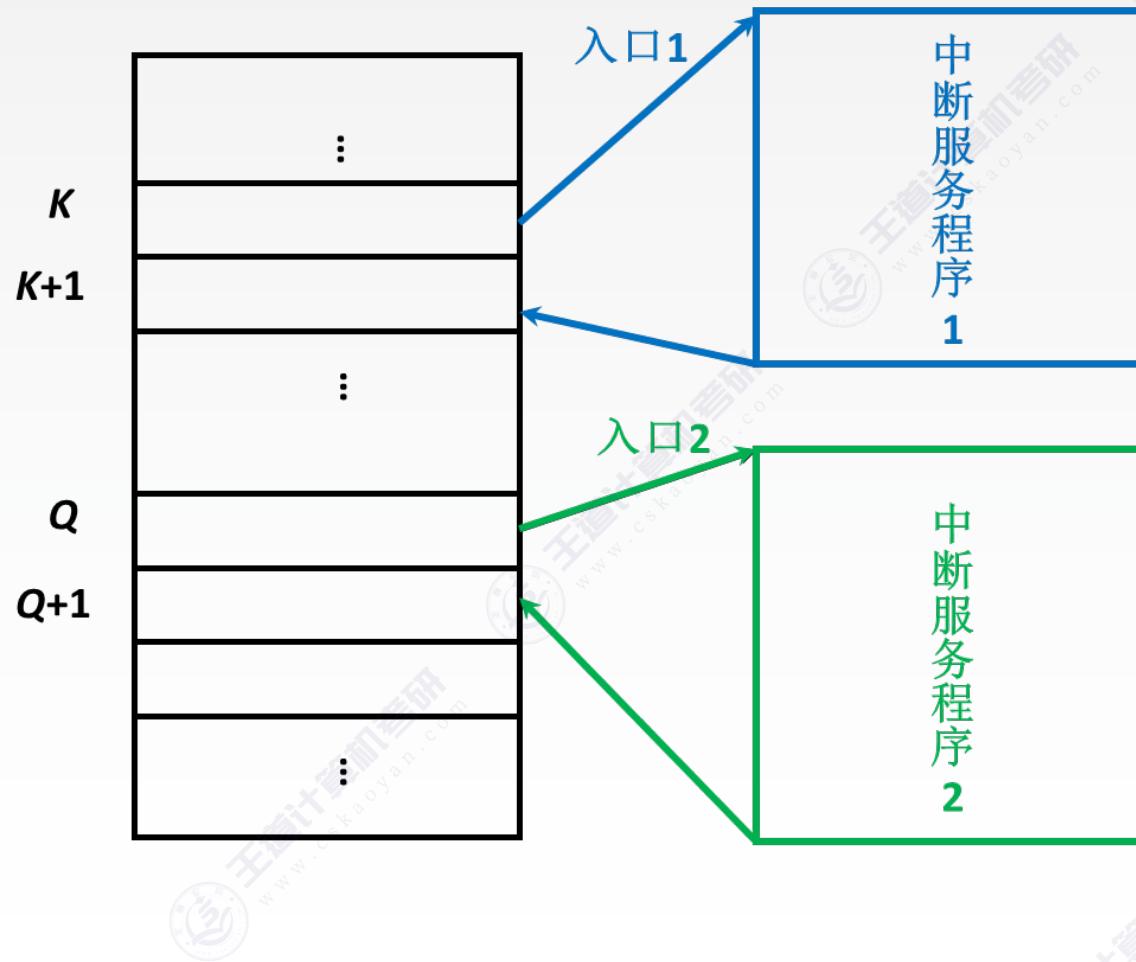
1. 硬件故障中断属于最高级，其次是软件中断；
2. 非屏蔽中断优于可屏蔽中断；
3. DMA请求优于I/O设备传送的中断请求
4. 高速设备优于低速设备；
5. 输入设备优于输出设备；
6. 实时设备优于普通设备。

中断处理过程



中断处理过程-中断隐指令

并不是一条具体的指令，而是CPU在检测到中断请求时自动完成的一系列动作



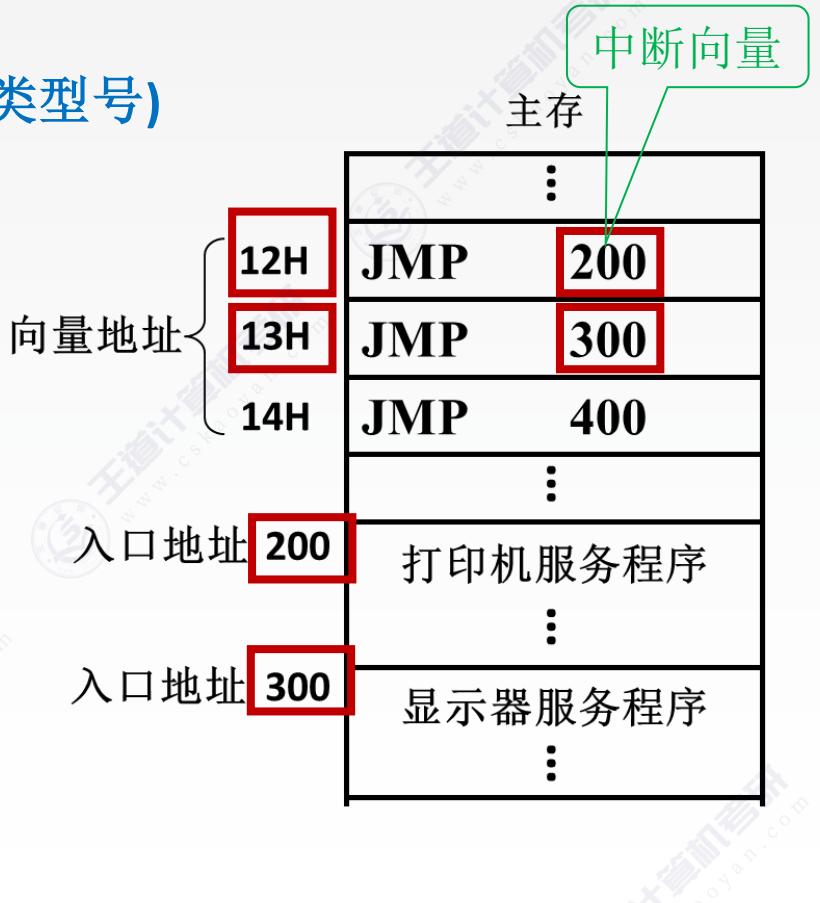
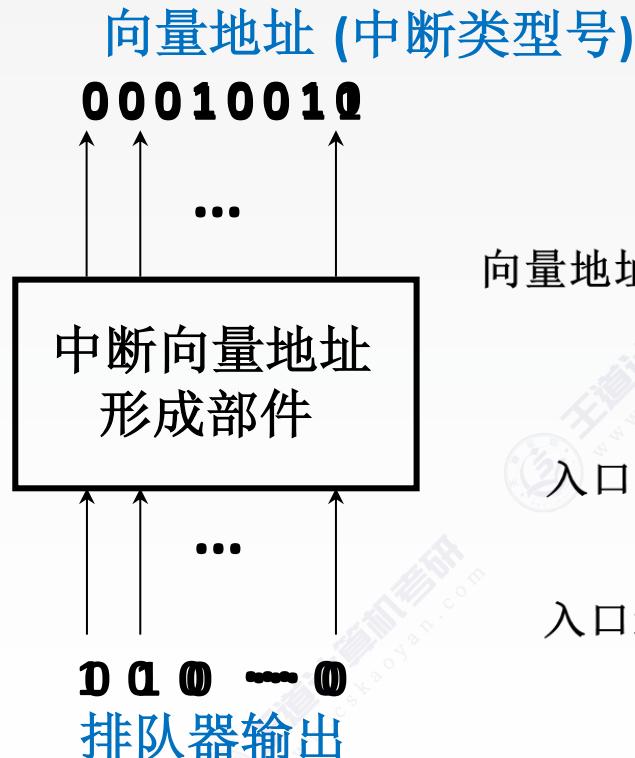
中断隐指令的主要任务：

- ① **关中断**。在中断服务程序中，为了保护中断现场（即CPU主要寄存器中的内容）期间不被新的中断所打断，必须关中断，从而保证被中断的程序在中断服务程序执行完毕之后能接着正确地执行下去。
- ② **保存断点**。为了保证在中断服务程序执行完毕后能正确地返回到原来的程序，必须将原来程序的断点（即程序计数器（PC）的内容）保存起来。可以存入堆栈，也可以存入指定单元。
- ③ **引出中断服务程序**。引出中断服务程序的实质就是取出中断服务程序的入口地址并传送给程序计数器（PC）。

{ 软件查询法
硬件向量法

中断处理过程-硬件向量法

由 **硬件** 产生 **向量地址**
再由 **向量地址** 找到 **入口地址**



中断隐指令的主要任务：

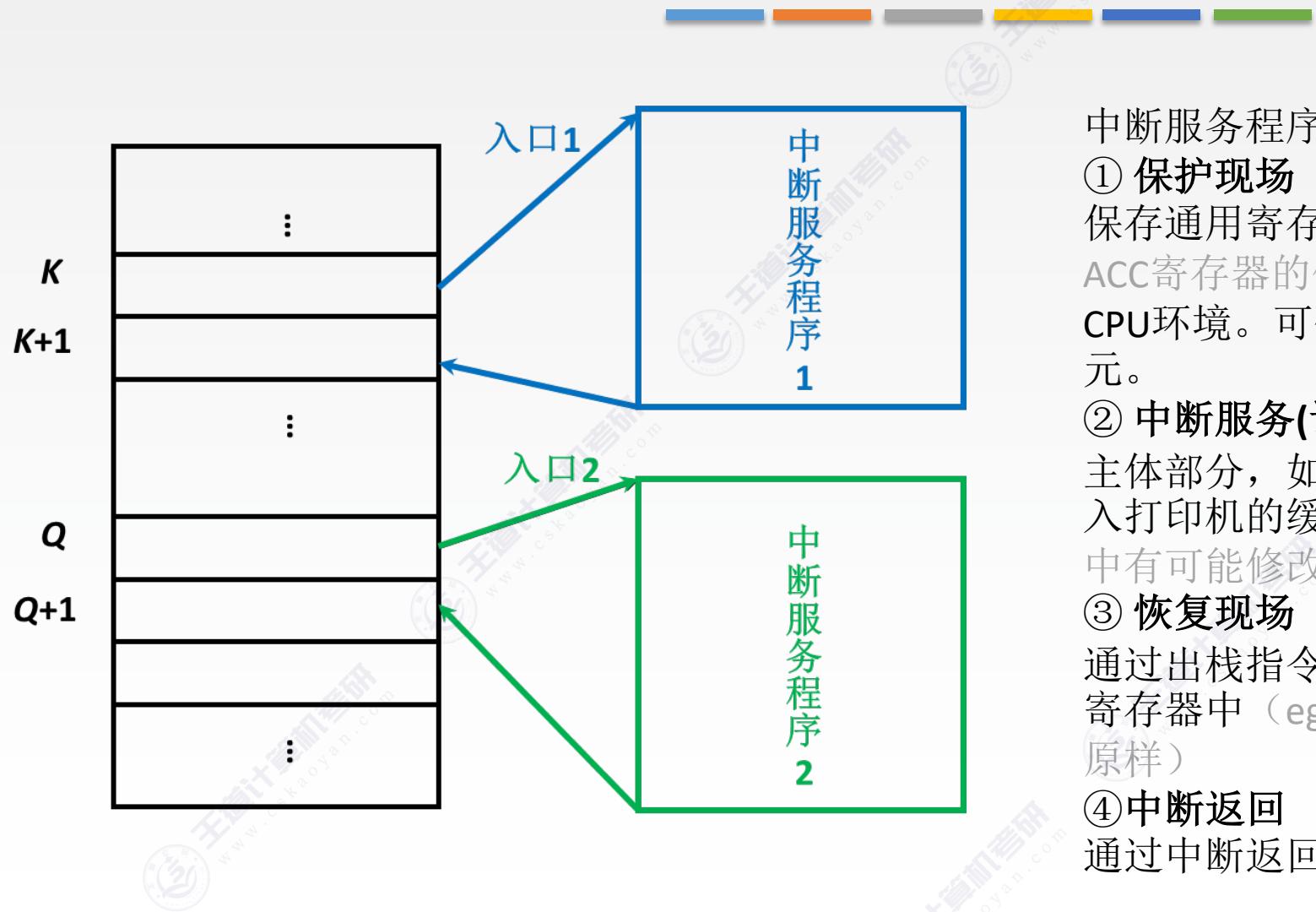
① **关中断**。在中断服务程序中，为了保护中断现场（即CPU主要寄存器中的内容）期间不被新的中断所打断，必须关中断，从而保证被中断的程序在中断服务程序执行完毕之后能接着正确地执行下去。

② **保存断点**。为了保证在中断服务程序执行完毕后能正确地返回到原来的程序，必须将原来程序的断点（即程序计数器（PC）的内容）保存起来。可以存入堆栈，也可以存入指定单元。

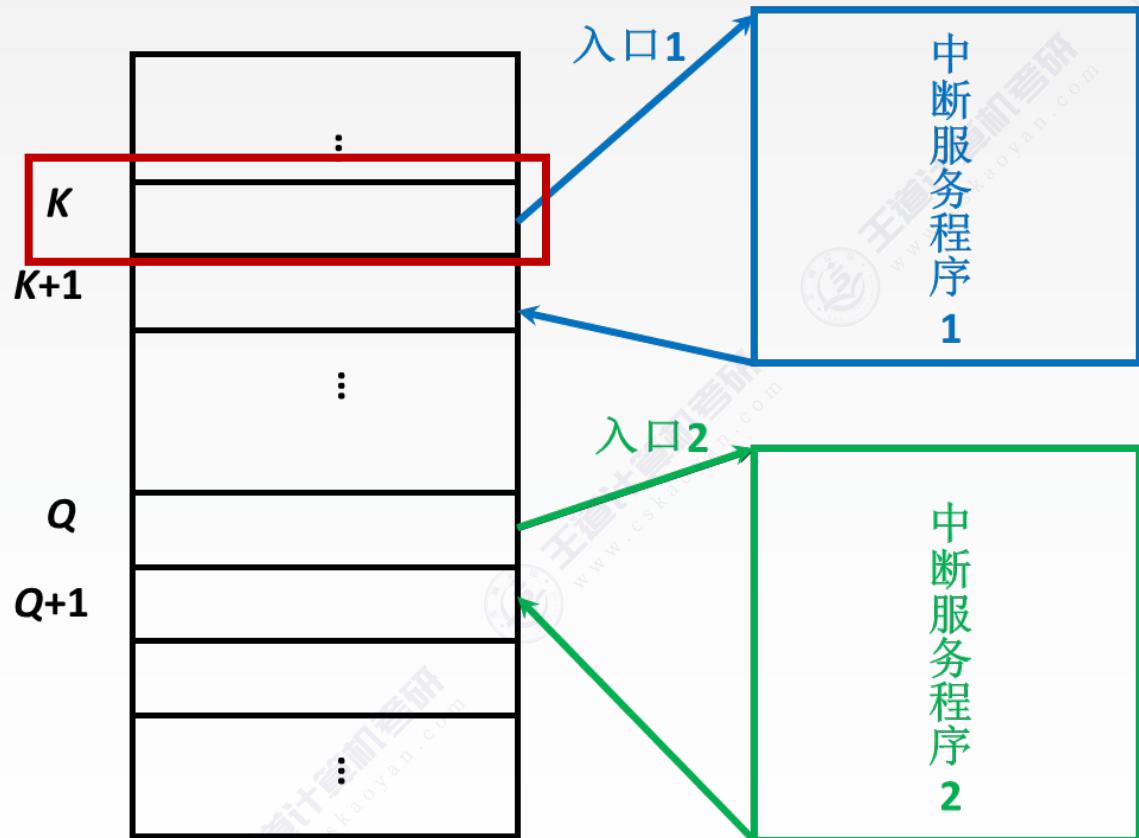
③ **引出中断服务程序**。引出中断服务程序的实质就是取出中断服务程序的入口地址并传送给程序计数器（PC）。

软件查询法
硬件向量法

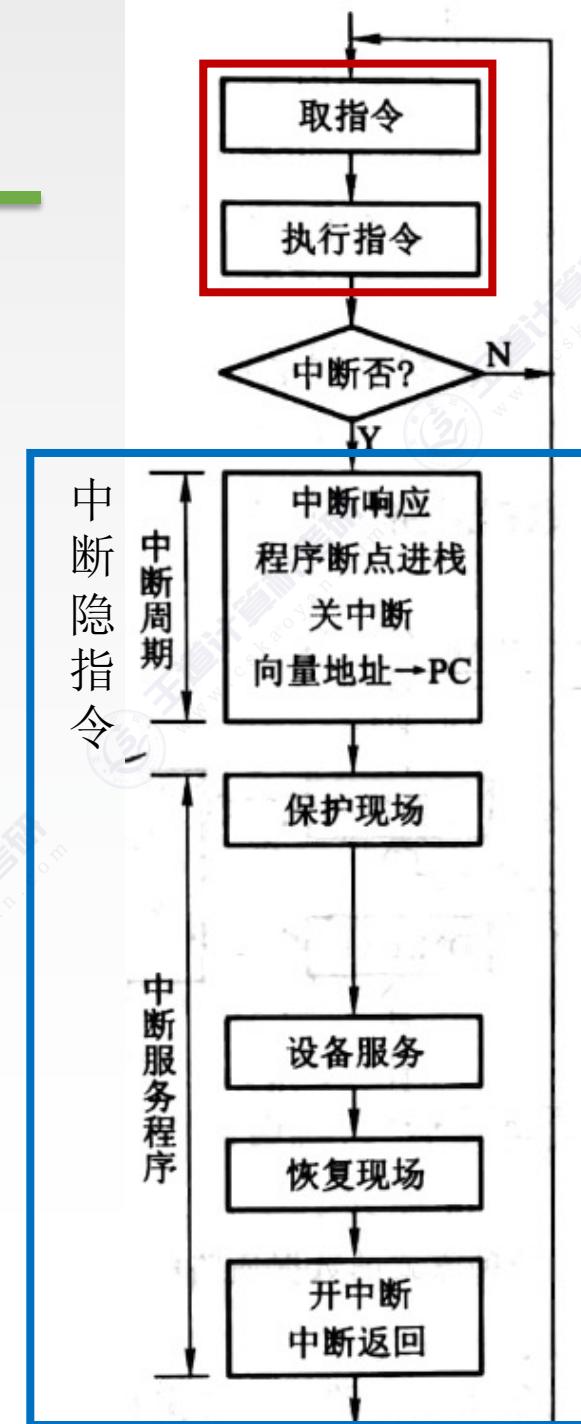
中断处理过程-中断服务程序



总结：中断处理过程



单重重断：执行中断服务程序时不响应新的中断请求。



中断处理过程