

Lecture 24: 知识 - 中断

中断基本概念

概念： 中断基本概念

- 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；
 - CPU 在每执行一条指令后，都需要看是否有中断处理请求，如果有的话就转去执行中断请求；
 - 当处理完中断处理程序后，再返回原本 PC 指向的位置、继续指向；
- 功能：
 - (1) 实现 CPU 与 I/O 设备的并行工作；
 - (2) 处理硬件故障和软件错误；
 - (3) 实现人机交互，用户干预机器需要用到中断系统；
 - (4) 实现多道程序、分时操作；
 - 多道程序的切换需借助于中断系统实时处理需要借助中断系统来实现快速响应；
 - (5) 实现应用程序和操作系统 (管态程序) 的切换，称为软中断；
 - (6) 多处理器系统中各处理器之间的信息交流和任务切换；
- 补充：一种理解中断的方式
 - 可以将中断理解为强制插入的系统调用，并且此系统调用直接由硬件实现；
 - 在这过程中，硬件负责保存当前程序的断点，即 `mepc = pc`，将当前断点存储到内存中；
 - 更新当前的程序状态字，即 `mcause.Interrupt = 1`
 - 然后跳转当前 PC 的指针，进入到内核进程执行，即 `PC = (mtvec & ~0xf)`

强制“插入”的 syscall

- 中断
 - 保存 `mepc = PC`
 - 跳转 `PC = (mtvec & ~0xf)`
 - 更新 `mcause.Interrupt = 1`
- 系统调用 (ecall)
 - 保存 `mepc = PC`
 - 跳转 `PC = (mtvec & ~0xf)`
 - 更新 `mcause.Ecall = 1`
- “无论你现在在做什么，都去执行一下操作系统内核的代码吧”

概念： 中断类型分类 - 是否屏蔽

- (1) 不可屏蔽中断
 - 不可屏蔽中断 (NMI) 用来通知 CPU 发生了紧急事件，如电源掉电；
 - 不可屏蔽中断在关中断时也会被响应；
 - CPU 处于关中断状态时，不会响应可屏蔽中断，但必须响应 NMI 请求；
 - NMI 一旦发生，CPU 需要立即响应，因此 NMI 的优先级比可屏蔽中断的优先级高；
- (2) 可屏蔽中断
 - 关中断时不会被响应；
 - 可屏蔽中断请求信号要求 CPU 处于开中断状态，且中断优先级较高才能被 CPU 响应；

概念： 中断类型分类 - 由内/外部产生

- (1) 内部中断：
 - 又称作异常；
 - 内部中断的检查发生在当前指令的执行过程当中；
 - 真题：2015年
 - 内部中断的产生与当前执行的指令有关，并且其检测由 CPU 内部逻辑实现；
- (2) 外部中断：
 - 狭义的中断；
 - 外部中断的检查发生在当前指令的执行后；

中断过程

中断过程概述

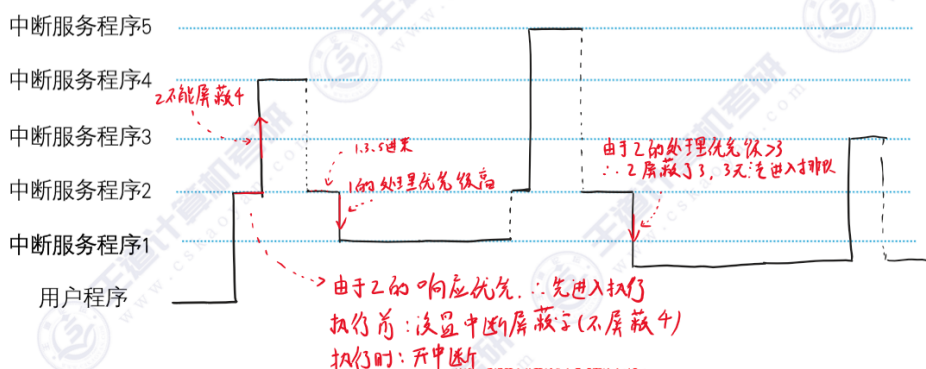
概念： 中断过程概述

- 第一步：中断申请（注意，不是中断响应）

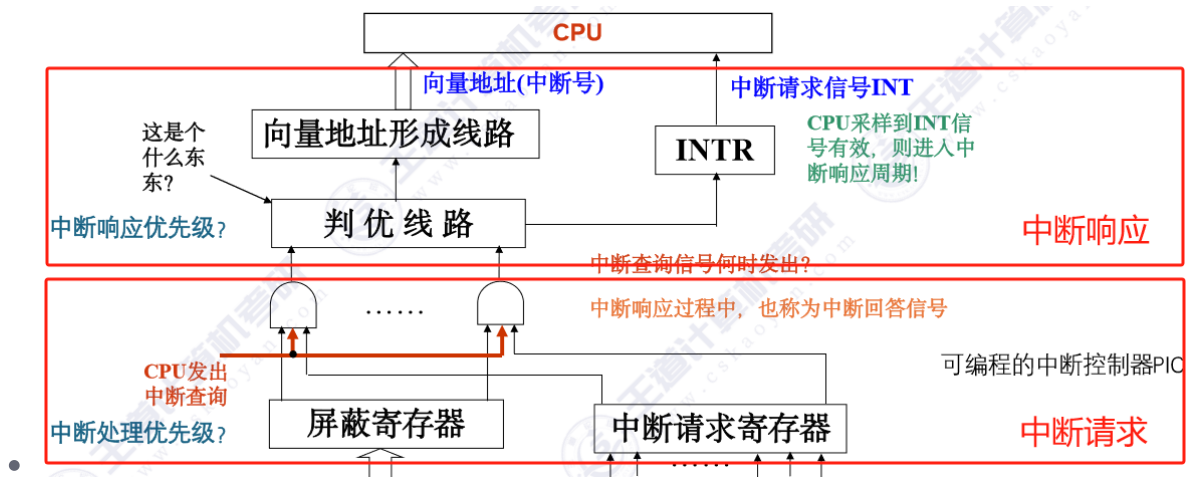
- (1) 中断请求
 - 中断源向 CPU 发送中断请求信号;
 - 把能够产生中断请求信号的设备称之为**中断源**;
 - 如果当前 CPU 已经处于关中断的状态, 则此时不响应中断;
- (2) 中断处理优先级 (能不能进 -> 屏蔽字)
 - 可以由**中断处理优先级**确定当前的**屏蔽字**, 基于**屏蔽寄存器**和中断请求寄存器, 决定当前是否可进入中断响应;
- (3) 中断响应优先级 (都能进的时候, 谁先进 -> 判优)
 - 真题: 2024年
 - 响应中断的条件: **中断判优** -> 多个中断源同时进入中断响应时, 通过中断判优逻辑响应一个中断源;
 - 接下来就可以知道要执行的中断服务程序;
 - 中断判优可以由中断判优线路实现, 也可以由软件实现;
- 总结:
 - 概念:
 - 中断处理优先级 -> 中断屏蔽字 -> 屏蔽: 决定能不能进;
 - 中断响应优先级 -> 判优线路 -> 响应: 决定谁先进;
 - 图示:
 - 当中断源 2、4 同时发出中断请求时, 因为 2 的中断响应优先级大于 4, 因此 2 先进入运行;
 - 但又因为 4 的中断处理优先级更高, 因此当 2 进入运行后, 无法屏蔽 4, 因此 4 进入运行、2 的断点被保存;

若某计算机中断系统有5个中断源, 分别记为1、2、3、4和5, 其中断响应优先级为 $1>2>3>4>5$, 而中断处理优先级为 $1>4>5>2>3$ 。要求:

(2) 若在运行用户程序时, 中断源2和4同时发生中断请求, 而在处理2号中断源的过程中, 中断源1、3和5同时发生中断请求, 试说明此时CPU运行的具体情况, 即响应和处理各个中断源的顺序。



- 图示:



• 第二步：中断处理

• (1) 中断隐指令 (也称之为 中断响应)

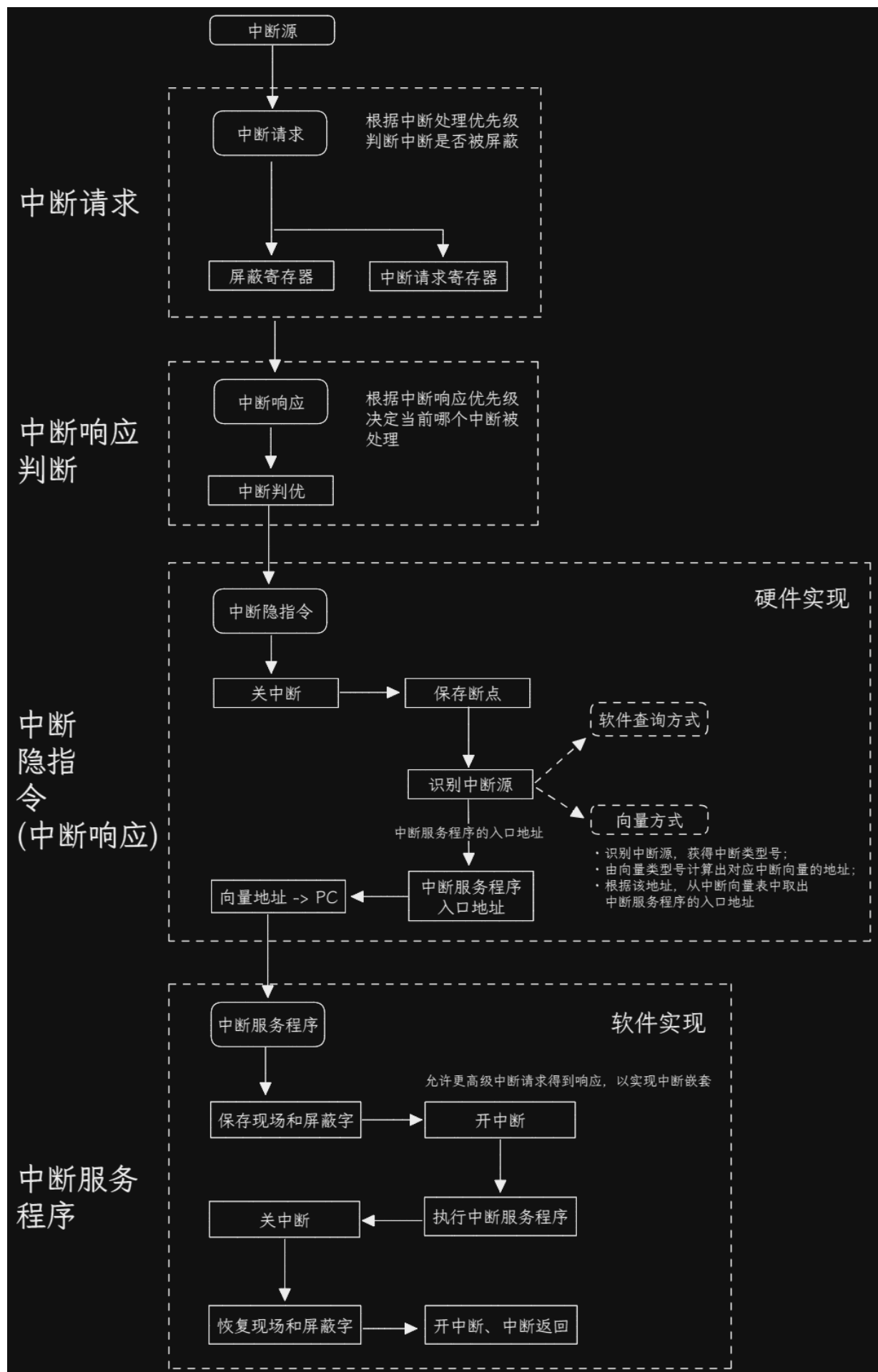
- 1. 关中断;
- 2. 保存断点;
- 3. 识别中断源;
- 4. 中断服务程序寻址;

• (2) 中断服务程序

- 1. 保存现场和屏蔽字;
 - 进入中断服务程序后首先要保存现场和中断屏蔽字;
 - 现场信息是指用户可见的工作寄存器的内容, 它存放程序执行到断点处的现行值;
 - 因为中断服务程序本质上是一串程序代码, 因此执行中断程序时, 类似于上下文切换、需要将当前正在执行的程序流的上下文保存;
- 2. 开中断 (可选);
 - 允许更高级中断请求得到响应, 以实现中断嵌套;
 - 注意: 如果是单级中断, 则不需要这一步;
- 3. 执行中断服务程序 (叫中断处理、中断事件处理都可以);
 - 这是中断请求的目的;
 - 真题: 2017年
 - 如果是多重中断时, 执行中断服务程序是在开中断下;
- 4. 关中断 (可选);
 - 保证在恢复现场和屏蔽字时不被中断;
 - 注意: 如果是单级中断, 则不需要这一步;
- 5. 恢复现场和屏蔽字;
 - 将现场和屏蔽字恢复到原来的状态;
 - 真题: 2024年
 - 因此需要设置新屏蔽字, 这一步由**软件实现**;
- 6. 开中断;
 - 中断服务程序的最后一条指令通常是一条中断返回指令, 使其返回到原程序的断点处, 以便继续执行原程序;

- 7. 中断返回;
- 注意:
 - 1. 中断隐指令 -> 硬件实现;
 - 2. 中断服务程序 -> 软件实现;
- 图示:

获取笔记持续更新可加微信: WandD2277



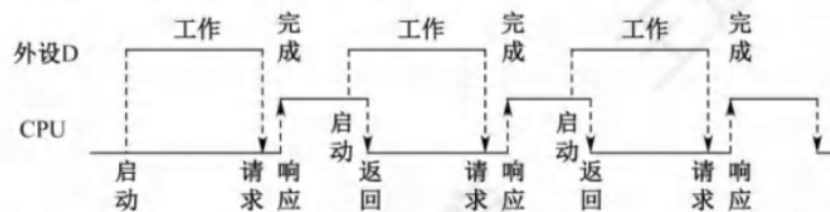
分析：中断过程分析

- 真题：2016年

• 题目：

【2016 统考真题】假定 CPU 主频为 50MHz，CPI 为 4。设备 D 采用异步串行通信方式向主机传送 7 位 ASCII 码字符，通信规程中有 1 位奇校验位和 1 位停止位，从 D 接收启动命令到字符送入 I/O 端口需要 0.5ms。回答下列问题，要求说明理由。

- 1) 每传送一个字符，在异步串行通信线上共需传输多少位？在设备 D 持续工作过程中，每秒最多可向 I/O 端口送入多少个字符？
- 2) 设备 D 采用中断方式进行输入/输出，示意图如下所示：



I/O 端口每收到一个字符申请一次中断，中断响应需 10 个时钟周期，中断服务程序共有 20 条指令，其中第 15 条指令启动 D 工作。若 CPU 需从 D 读取 1000 个字符，则完成这一任务所需时间大约是多少个时钟周期？CPU 用于完成这一任务的时间大约是多少个时钟周期？在中断响应阶段 CPU 进行了哪些操作？

• 解析：

• 第一问：

- 异步串行通信线上需要传输位有：

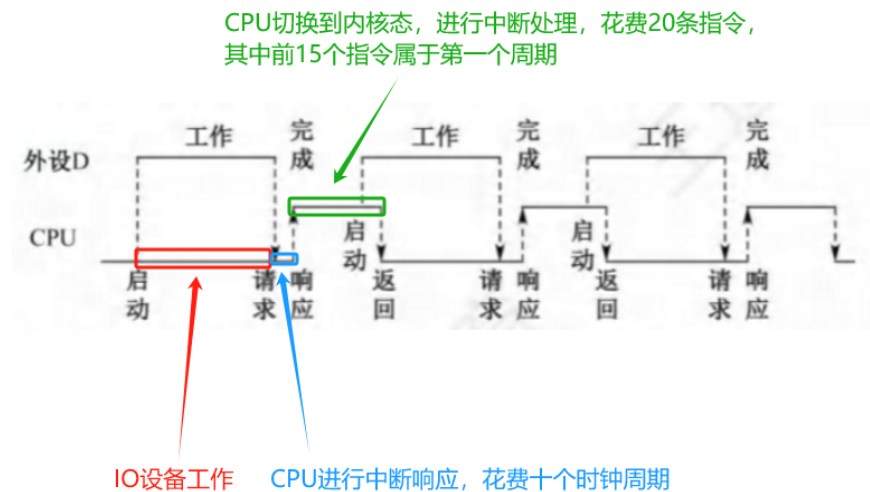
- 1. 起始位 (1 位)；
- 2. 数据位 (7 位)；
- 3. 奇偶校验位 (1 位)
- 4. 停止位 (1 位)；
- 因此总位数为 10 位；

- 且已知当前每传送一个字符的数据需要 0.5ms，因此在 1s 内，一共可以送入 $1s/0.5ms=2000$ 个，即可以送入 2000 个字符；

• 第二问：

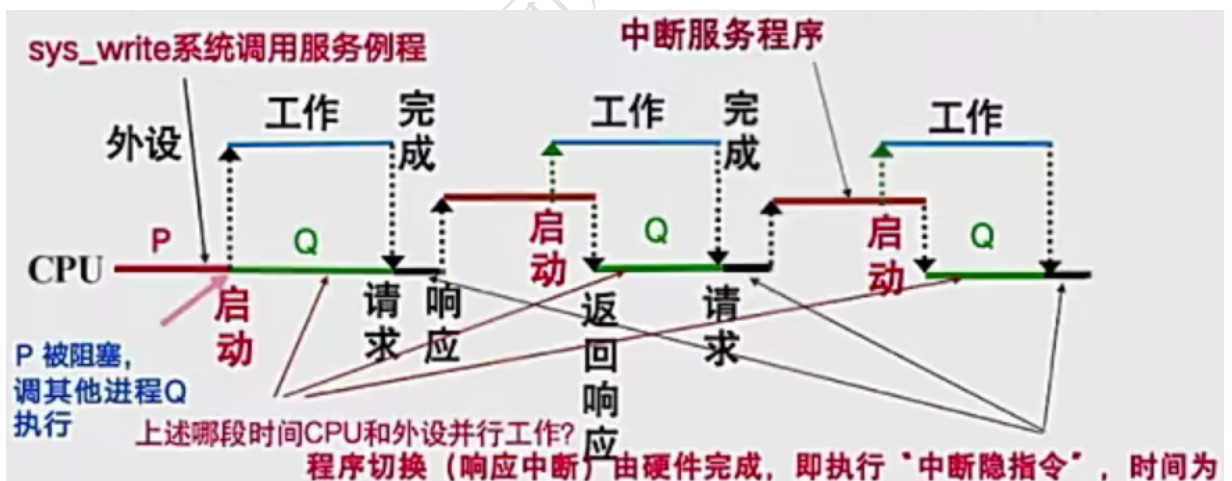
- 由图可知，设备 D 的一次字符 I/O 任务都包括以下流程：

- 1. CPU 启动设备 IO，IO 开始工作；
- 2. IO 设备消耗 0.5ms，将数据读至 IO 控制器的缓冲区；
- 3. IO 控制器发出中断请求，CPU 消耗 10 个时钟周期进行中断响应；
- 4. CPU 执行中断服务程序，执行中断服务程序的 20 条指令；
 - 由于在执行中断服务程序的第 15 条指令时 IO 设备就继续开始了工作，因此一个周期的 IO 任务可以理解为只需执行 15 条中断服务程序；
- 如图所示：



- 综上, 将字符送到 IO 端口所需的时间、中断响应时间、中断服务程序执行前的 15 条指令启动 D 工作的时间为一个字符的传送时间, 即: $15 \times 4 + 10 + 0.5\text{ms} \times 50\text{M/s} = 25070$ 个时钟周期;
 - 因此 1000 个字符的传送所需时间为 25070000 个时钟周期;
- 由上可得, 中断服务程序共有 20 条指令, CPI 为 4, 中断响应为 10 个时钟周期, 得到 CPU 用于完成这一任务的时间约为 $1000 \times (10 + 20 \times 4) = 90000$ 个时钟周期;
- 在中断响应阶段, CPU 主要进行以下操作:
 - 关中断、保存断点和程序状态、识别中断源, 获得中断服务程序地址;

• 图示:



中断申请

中断请求与中断响应优先级

概念: 中断请求

- 别称
 - (是否接受) 中断响应
- 概念: 中断请求
 - 当程序执行第 i 条指令时, 若检测到一个异常事件 (内中断, 在指令执行当中), 或在执行第 i 条指令后, 发现有一个中断请求信号 (外中断, 在指令执行之后), 则 CPU 会中

断当前程序的执行，跳转到操作系统中相应的 异常/中断处理程序 去执行；

- 若 异常/中断处理程序 能够解决相应问题，则在 异常/中断处理程序 的最后，CPU 通过执行“异常/中断返回指令”回到被打断的用户进程的第 i 条指令或第 $i+1$ 条指令继续执行；

- 真题：2023年

- 错误选项：外部设备通过中断控制器向 CPU 发中断结束信号；

- 这个选项是错误的。CPU 通过执行“异常/中断返回指令”返回原本的程序流；

- 若 异常/中断处理程序 发现是不可恢复的致命错误，则**终止用户进程**；

- 通常情况下，对于异常和中断事件的具体处理过程全部由操作系统软件来完成；

- 中断的**检测和响应**由硬件完成，中断的具体处理由操作系统软件完成；

- 一旦 CPU 检测到有异常事件或中断请求，就会进入异常/中断响应过程，在此过程中 CPU 完成以下两个任务：保护断点和程序状态、识别异常/中断类型并转相应处理程序执行；

- 概念：标记触发器

- 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 INTR 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；

- 总结：中断响应时刻：

- 内中断：

- 对于内中断（异常），CPU 在执行一条**指令的过程中**检测异常事件。CPU 执行每条指令的过程中，都会对异常事件和中断请求进行检测；

- 例如，在指令译码时，会检测指令操作码是否合法；

- 在取指令或取数据时，会检测是否发生了缺页或访问越权等存储保护错；

- 执行整数除法指令时，会检测除数是否为 0；

- 在每条指令执行结束时；会检测有没有中断请求；

- 外中断：

- 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 IO 的中断请求；

- 也就是说，外中断下，CPU 响应中断的时间是在每条指令的**执行阶段的结束时刻**；

- 总结：CPU 响应中断条件

- 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；

- 真题：2023年

- 在开中断下，CPU 检测到中断请求后、就进行中断响应；

- 注意其不会被屏蔽，因为当前并没有处于中断下，所以不会被屏蔽；

- 图示：

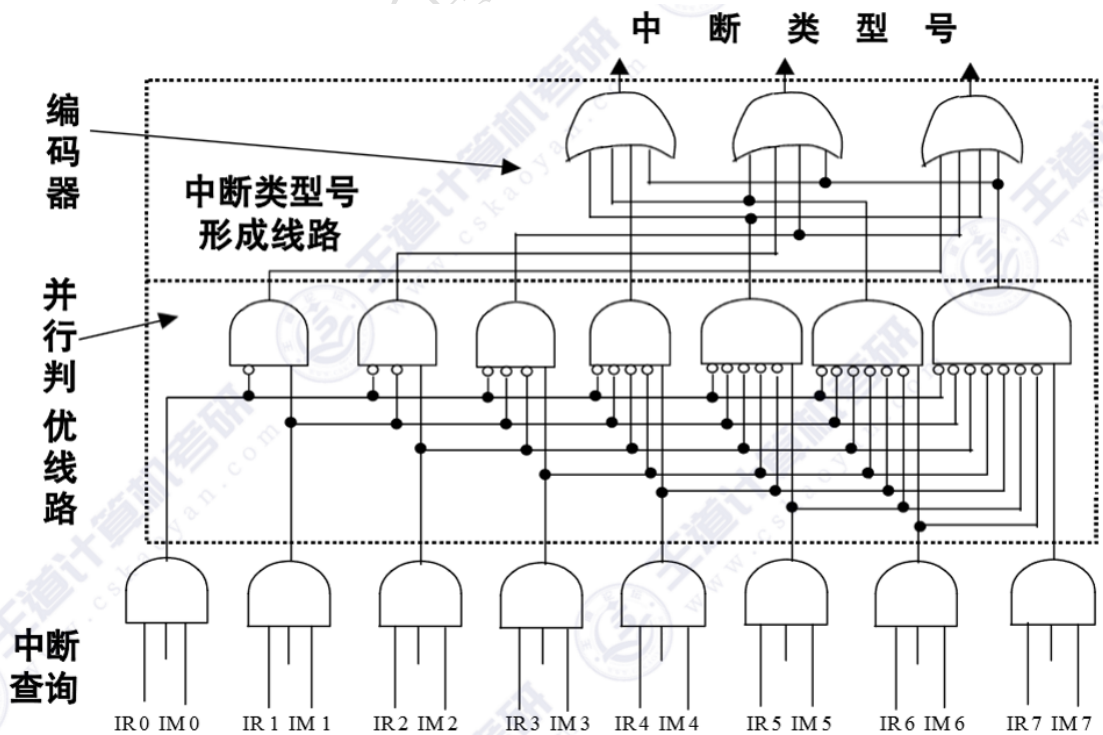
- 中断请求标记寄存器

中断请求标志寄存器	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

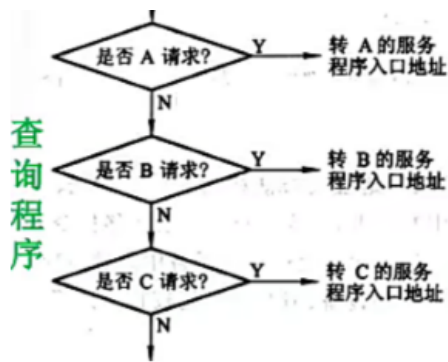
中断处理优先级与中断判优

概念：中断处理优先级与中断判优

- 概念：
 - 中断屏蔽字是决定中断处理优先级；
 - 中断判优**用于多个中断源同时进入中断响应时，通过中断判优逻辑选择一个中断源进行处理；
 - 中断判优既可以用硬件实现，也可用软件实现；
- 注意：
 - 真题：**2021年
 - 进入中断响应周期时，**CPU** 一定处于开中断状态；
 - 若 **CPU** 检测到中断请求信号，则一定存在**未被屏蔽的中断源请求信号**
- 硬件实现：
 - 概念：
 - 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；
 - 图示：



- 软件实现：
 - 概念：
 - 是通过**查询程序**实现的；
 - 图示：



• 优先级设置：

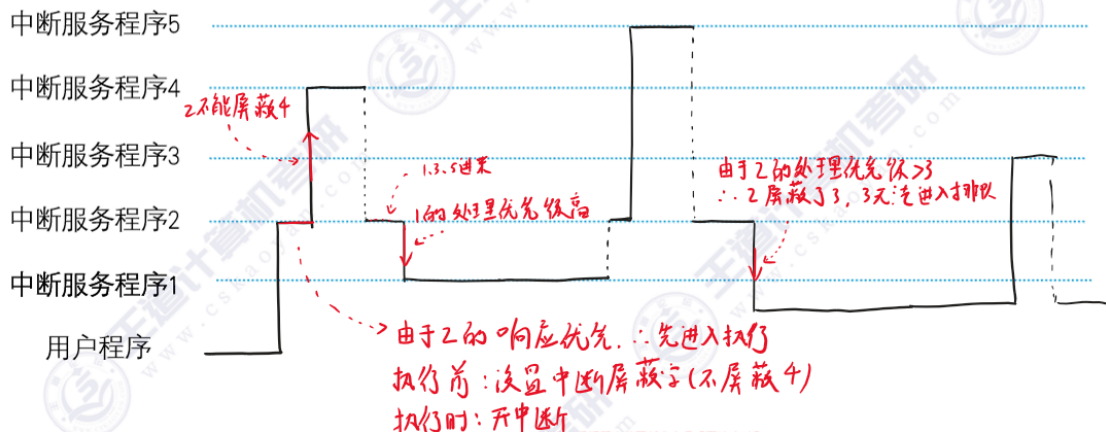
1. 硬件故障中断属于最高级，其次是软件中断；
2. 非屏蔽中断优于可屏蔽中断；
3. DMA请求优于I/O设备传送的中断请求
4. 高速设备优于低速设备；
5. 输入设备优于输出设备；
6. 实时设备优于普通设备。

分析：中断响应与处理

- 当中断源 2、4 同时发出中断请求时，因为 2 的中断响应优先级大于 4，因此 2 先进入运行；
- 但又因为 4 的中断处理优先级更高，因此当 2 进入运行后，无法屏蔽 4，因此 4 进入运行、2 的断点被保存；
- 4 运行完成后，中断嵌套返回，2 继续运行。2 运行时 1、3、5 同时请求，因为 1 的响应优先级更高，先响应 1 的请求；
- 因为 1 的处理优先级比 2 高，2 无法屏蔽 1，因此 2 的断点被保存、1 运行中断服务程序；
- 1 运行完成后，返回 2，此时 3 和 5 进入中断响应，虽然 3 可以先被响应，但是因为其被 2 屏蔽，因此无法运行。接着响应 5，因为 2 无法屏蔽 5，因此 5 的中断服务程序运行，后续同理；

若某计算机中断系统有5个中断源,分别记为1、2、3、4和5,其中断响应优先级为 $1>2>3>4>5$,而中断处理优先级为 $1>4>5>2>3$ 。要求:

(2)若在运行用户程序时,中断源2和4同时发生中断请求,而在处理2号中断源的过程中,中断源1、3和5同时发生中断请求,试说明此时CPU运行的具体情况,即响应和处理各个中断源的顺序。



中断处理

中断隐指令

概念: 中断隐指令

- 1.1 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;
- 1.2 保存断点:
 - 概念:
 - 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;
 - 如果是支持多重中断的系统, 只能将 PC 值存入栈中。如果是单重中断, 则都可以;
 - 分析: 保存 PC 分析
 - (1) 如果是 trap 指令, 则保存的 PC 应当为自陷指令的后一条指令的地址;
 - (2) 如果是缺页导致的中断, 则保存的 PC 应当是导致缺页的那一条 Load/Use 指令的地址, 因此应当是 $PC-4$
 - 因为执行 Load/Use 指令后、会进行 $PC+1$, 假设是 32 位机器, 则是 $PC+4$, 因此应当保存 $PC-4$ 的值;
 - 注意:
 - 保存断点, 即 PC 的值是中断隐指令完成的, 而保存屏蔽字、上下文环境, 则是由中断服务程序完成的, 注意它们的区别;
- 1.3 识别中断源
 - 概念:
 - 中断识别分为向量中断和非向量中断两种, 非向量中断即软件查询法;
 - 每个中断源都有一个唯一的类型号, 每个中断类型号都对应一个中断服务程序, 每个中断服务程序都有一个入口地址, 即中断向量, CPU 必须找到入口地址;

- 把系统中的全部中断向量集中存放到存储器的某个区域内，这个存放中断向量的存储区就称为中断向量表；
- CPU 响应中断后，通过识别中断源获得中断类型号，然后据此计算出对应中断向量的地址；
- 再根据该地址从中断向量表中取出中断服务程序的入口地址，并送入程序计数器 PC，以转去执行中断服务程序，这种方法被称为中断向量法，采用中断向量法的中断被称为向量中断；

- 流程：

- 识别中断源 -> 获得中断类型号 -> 计算中断向量地址 -> 查中断向量表，取得中断服务程序入口地址；

- 1.4 引出中断服务程序：

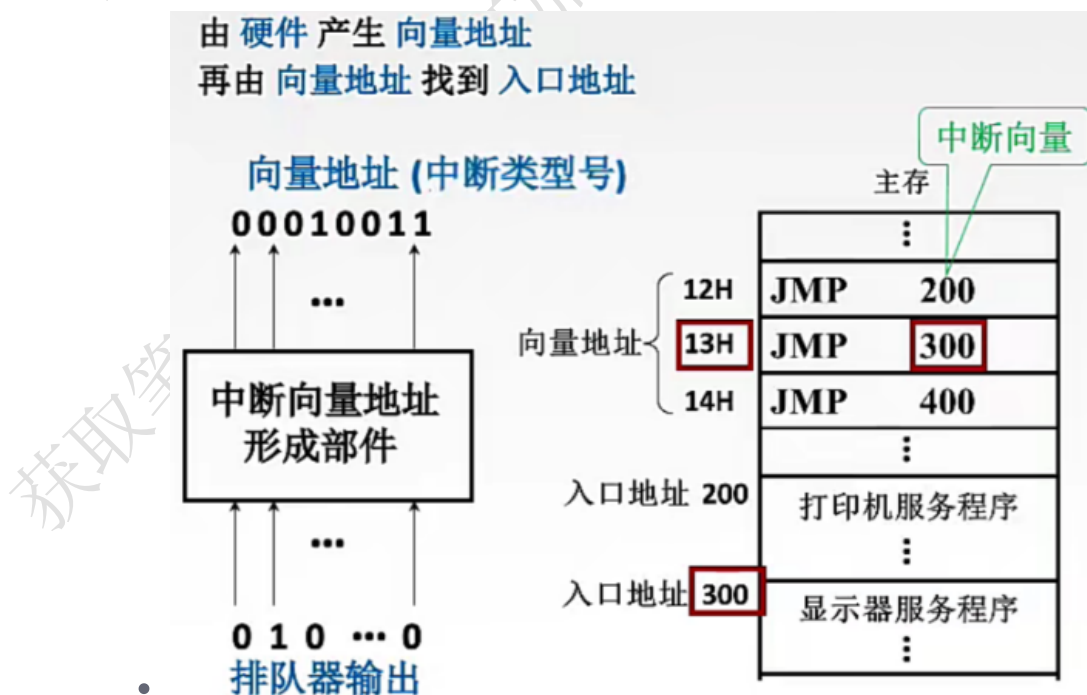
- 方式：

- 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)；
- 一般来说，内部异常事件和外部中断源的识别方式不同，大多数处理器会将两者分开来处理；
- 内部异常事件的识别大多采用软件识别方式，而外部中断源则可以采用软件识别或硬件识别方式；

- 分类：

- 硬件识别

- 图示：



- 概念：

- 由于外部中断的发生与 CPU 正在执行的指令没有必然联系，相对于指令来说，外部中断是随机的、与当前执行指令无关，所以，并不能根据指令执行过程中的某些现象来判断是否发生了中断请求；
- 因此，对于外部中断，只能在每条指令执行完后、取下条指令之前去查询是否有中断请求；

- 通常 CPU 通过采样对应的**中断请求引脚**来进行查询，如果发现中断请求信号有效，则说明有中断请求，但是到底是哪个中断源发出的请求，还需要进一步识别有关外部中断源的响应、识别和中断处理程序的结构等内容；
- 软件识别
 - 软件识别方式是指，CPU 设置一个异常状态寄存器，用于记录异常原因；
 - 操作系统使用一个统一的异常查询程序，该程序按一定的优先级顺序查询异常状态寄存器，先查询到的异常先被处理；
 - 例如，MIPS 就采用软件识别方式，CPU 中有一个 **cause** 寄存器，位于地址 **0x80000180** 处有一个专门的异常查询程序，它通过查询 **cause** 寄存器来检测异常类型，然后转到内核中相应的异常处理程序进行具体的处理；
 - 因为像故障和陷阱之类的内部异常通常是在执行某条指令时发现的，可以通过对指令执行过程中某些条件的判断来发现是否发生了异常，而且一旦发现可以马上进行处理；
 - 所以，内部异常事件也可以不通过专门的查询程序来识别，而在发现异常时直接得到异常类型，根据类型号转到相应的异常处理程序执行；
 - IA-32 的处理方式就是这样；

概念：中断向量、向量地址、向量中断

- 中断向量：
 - 每个中断源都有对应的处理程序，这个处理程序称为中断服务程序，其入口地址称为中断向量；
 - 所有中断的中断服务程序入口地址构成一个表，称为**中断向量表**；
 - 也有的机器把中断服务程序入口的跳转指令构成一张表，称为**中断向量跳转表**；
- 向量地址：
 - 中断向量表或中断向量跳转表中每个表项所在的内存地址或表项的索引值，称为向量地址或**中断类型号**；
- 向量中断：
 - 指一种识别中断源的技术或方式；
 - 识别中断源的目的是找到中断源对应的中断服务程序的入口地址的地址，即获得向量地址；

中断服务程序

概念：中断服务程序

- 单级中断：
 - **真题**：2010年
 - 1. 保存现场和屏蔽字；
 - 2. 执行中断服务程序；
 - 3. 恢复现场；
 - 4. 开中断
 - 5. 中断返回；

- 多级中断：
 - 1. 保存现场和屏蔽字；
 - 进入中断服务程序后首先要保存现场和中断屏蔽字，现场信息是指用户可见的工作寄存器的内容，它存放程序执行到断点处的现行值；
 - 保存**通用寄存器**和**状态寄存器**的内容（比如保存 **ACC** 寄存器的值）；
 - 以便返回原程序后可以恢复 **CPU** 环境；
 - 可使用堆栈，也可以使用特定存储单元；
 - 2. 开中断；
 - 允许更高级中断请求得到响应，以实现中断嵌套；
 - 3. 执行中断服务程序；
 - 这是中断请求的目的；
 - 4. 关中断；
 - 保证在恢复现场和屏蔽字时不被中断；
 - 5. 恢复现场和屏蔽字；
 - 将现场和屏蔽字恢复到原来的状态；
 - 通过出栈指令或取数指令把之前保存的信息送回寄存器中；
 - 6. 开中断；
 - 7. 中断返回；
 - 中断服务程序的最后一条指令通常是一条中断返回指令，使其返回到原程序的断点处，以便继续执行原程序；
- 总结：
 - 多级中断和单级中断相比，主要是**多了一对关中断和开中断**的操作。由于多级中断可能会发生嵌套中断，因此要保证在恢复现场和屏蔽字时不被中断，所以为其关中断；

多重中断

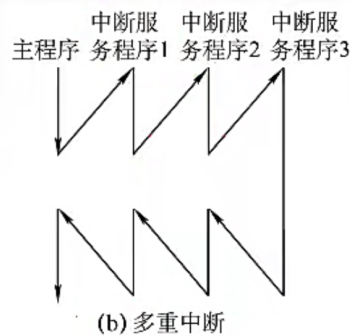
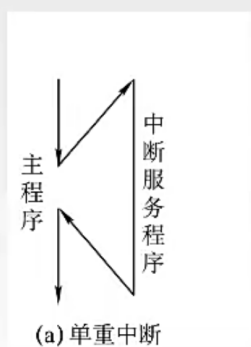
多重中断概念

概念：单重中断

- 指执行中断服务程序时，不响应新的中断请求；

概念：多重中断

- 概念：
 - 又称中断嵌套，执行中断服务程序时可响应新的中断请求；
 - 即可以在中断服务程序执行时、嵌套其他的中断服务程序；
- 注意：
 - 支持多重中断的系统中，需要**先保护完现场以及屏蔽字**之后、才可以开中断，并运行执行中断服务程序；
 - 在一条指令结束时响应中断；
- 图示：



	单重中断	多重中断
中断隐指令	关中断	关中断
	保存断点 (PC)	保存断点 (PC)
	送中断向量	送中断向量
中断服务程序	保护现场	保护现场和屏蔽字
	-	开中断
	执行中断服务程序	执行中断服务程序
	-	关中断
	恢复现场	恢复现场和屏蔽字
	开中断	开中断
	中断返回	中断返回

• 真题：2021年

• 题目：

【2021 统考真题】下列是关于多重中断系统中 CPU 响应中断的叙述，错误的是（ ）。

- A. 仅在用户态（执行用户程序）下，CPU 才能检测和响应中断
- B. CPU 只有在检测到中断请求信号后，才会进入中断响应周期
- C. 进入中断响应周期时，CPU 一定处于中断允许（开中断）状态
- D. 若 CPU 检测到中断请求信号，则一定存在未被屏蔽的中断源请求信号

• 答案：

- A 在内核态下，CPU 也可以检测和响应中断；

屏蔽字

概念：屏蔽字

• 概念：

- 当 CPU 来处理中断信号时，需要屏蔽字来决定：1) 哪些中断可以被响应，2) 它们的优先级；
- 中断屏蔽技术主要用于多重中断，CPU 要具备多重中断的功能，须满足下列条件：
 - 1. 在中断服务程序中提前设置开中断指令；
 - 2. 优先级别高的中断源有权中断优先级别低的中断源；

• 中断屏蔽级数：

- 每个中断源都有一个屏蔽触发器；
- 1 表示屏蔽该中断源的请求；
- 0 表示可以正常申请；
- 所有屏蔽触发器组合在一起，便构成一个屏蔽字寄存器，屏蔽字寄存器的内容称为屏蔽字；

• 屏蔽字设置规律：

- 1. 一般用' 1 '表示屏蔽, ' 0 '表示正常申请;
- 2. 每个中断源对应一个屏蔽字, 在处理该中断源的中断服务程序时, 屏蔽寄存器中的内容为该中断源对应的屏蔽字;
- 3. 屏蔽字中' 1 '越多, 优先级越高, 每个屏蔽字中至少有一个至少要能屏蔽自身的中断;

• 例题:

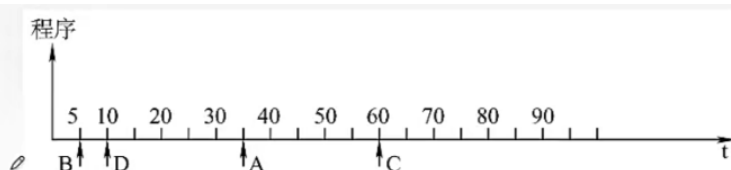
• 题目:

设某机有4个中断源A、B、C、D, 其硬件排队优先次序为A>B>C>D, 现要求将中断处理次序改为D>A>C>B。

1) 写出每个中断源对应的屏蔽字。

2) 按下图所示的时间轴给出的4个中断源的请求时刻, 画出CPU执行程序的轨迹。设每个中断源的中断服务程序时间均为20us。

• 第一问:



1)

中 断 源	屏 蔽 字			
	A	B	C	D
A	1	1	1	0
B	0	1	0	0
C	0	1	1	0
D	1	1	1	1

中断源A的屏蔽字为1110

中断源B的屏蔽字为0100

中断源C的屏蔽字为0110

中断源D的屏蔽字为1111

• 第二问:

2) 按下图所示的时间轴给出的4个中断源的请求时刻, 画出CPU执行程序的轨迹。设每个中断源的中断服务程序时间均为20us。

