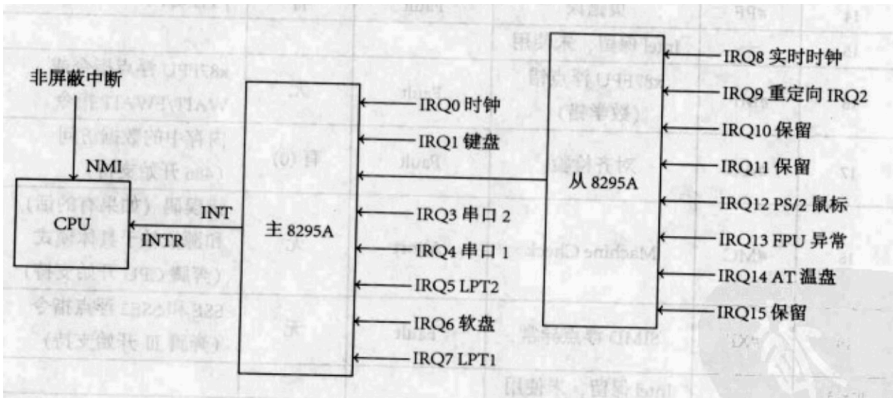
# 目 录

# 1 中断和异常的实现

## 1.1 设置 8259A

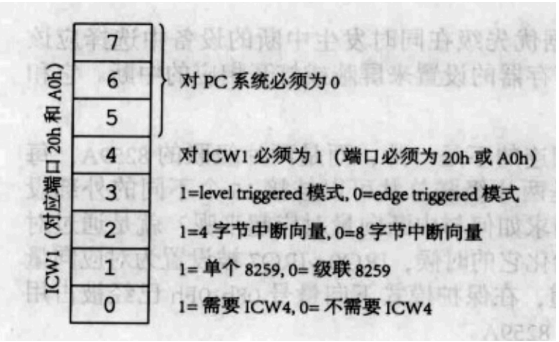
8259A 是中断机制中所有外围设备的一个代理，这个代理可以根据优先级在同时发生中断的设备中选择应该处理的请求。除此之外，还可以通过对 8259A 的寄存器的设置来屏蔽或打开相应的中断。可屏蔽外部中断与 CPU 是通过 8259A 连接起来的。8259A 与 CPU 的连接如下图所示：




由图可知，每一片 8259A 有 8 根中断信号线，两片级联的 8259A 可以挂接 15 个不同的外部设备。这些外部设备发出中断请求时，8259A 将其与相应的中断向量号对应起来。所以我们需要设置 8259A。

设置 8259A 的过程就是向其相应的端口写入特定的 ICW。主 8259A 的端口有 20h 和 21h，从 8259A 的端口有 A0h 和 A1h。ICW 全称是 Initialization Command Word，大小为一个字节。初始化 8259A 的过程如下：

- 首先往端口 20h 和 A0h 写入 ICW1。ICW1 的格式如图所示：




- 然后往端口 21h 和 A1h 写入 ICW2。主 8259A 和从 8259A 的 ICW2 内容可以不一样。写入 ICW2 时涉及与中断向量号的对应。比如，往主 8259A 写入 ICW2 时，如

# 2 保护模式下的 I/O

I/O 的控制权限是需要严格控制的，操作系统通过 IOPL 和 I/O 许可位图实现对 I/O 控制权限的限制。

## 2.1 IOPL

IOPL 字段位于 Eflags 寄存器的第 12、13 位。如下图所示：

操作系统将一些指令定义为 I/O 敏感指令，这些指令只有在 CPL≤IOPL 时才能执行，如果低特权级的任务试图执行这些指令将会引起一般性保护异常。I/O 敏感指令包括 in、ins、out、outs、cli 和 sti。

IOPL 字段是可以修改的，程序可以通过 popf 和 iretd 指令修改 IOPL 字段。只有当任务特权级为 0 时，popf 和 iretd 才可以成功修改 IOPL 的值。否则即使执行了指令，IOPL 也不会改变，不过也不会引起异常。

popf 指令还可以用来改变 IF 标志，只有当 CPL≤IOPL 时，才能成功修改 IF 标志，否则 IF 将维持原值，不会产生任何异常。

## 2.2 I/O 许可位图

在第五次学习报告中，我有实现过 TSS。其中代码有一处是“I/O 位图基址”。I/O 位图基址指向的就是 I/O 许可位图。I/O 许可位图的每一位用于表示一个字节的端口地址是否可用。如果该位为 0，表示此位对应的端口号可用，为 1 则代表不可用。I/O 许可位图的使用使得即便在同一特权级下不同任务也可以有不同的 I/O 访问权限。

I/O 许可位图就位于 TSS 段中，而 I/O 位图基址实际上是以 TSS 的地址为基址的偏移。如果 I/O 位图基址大等于 TSS 段界限，就表示 TSS 段中没有 I/O 许可位图。由于每个任务都有单独的 TSS，所以每个任务都有自己单独的 I/O 许可位图。

下面是一个任务中 I/O 许可位图的实现代码：

```
[SECTION .tss3]
LABEL_TSS3:
    DW $ - LABEL_TSS3 + 2 ; 指向I/O许可位图
times 12 DB 0FFh ; 端口00h~5fh都不可用
    DB 11111101b ; 端口60h~67h，只有端口61h可以用
    DB 0FFh ; I/O许可位图结束标志,I/O许可位图必须以0FFh结尾
TSS3Len equ $ - LABEL_TSS3
```

# 3 linux 下的内存管理