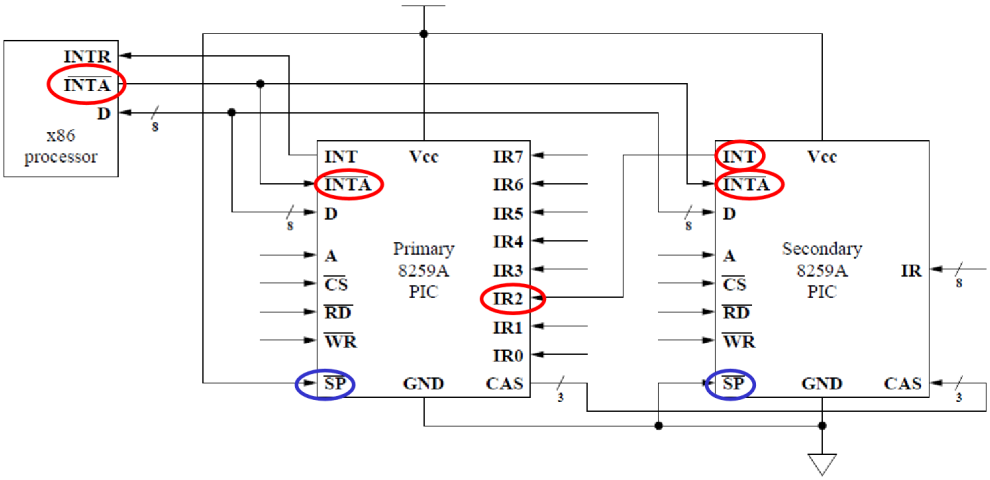
	mutual exclusion	reader/writer	
data shared by interrupt handlers	`spin_lock_irqsave` `spin_unlock_irqrestore`	`read_lock_irqsave` `read_unlock_irqrestore`	`write_lock_irqsave` `write_unlock_irqrestore`
data shared only by system calls	`down` `up`	`down_read` `up_read`	`down_write` `up_write`

type of code entering critical section	critical section shares data with	for mutual exclusion, use
system calls only	other system calls	`up` `down`
	interrupt handlers	`spin_lock_irq` `spin_unlock_irq`
both system calls and interrupt handlers	both system calls and interrupt handlers	`spin_lock_irqsave` `spin_unlock_irqrestore`
interrupt handlers only	system calls	`spin_lock` `spin_unlock`
	higher priority interrupt handlers	`spin_lock_irqsave` `spin_unlock_irqrestore`

```
/*
 * outb_p - this has to work on a wide range of PC hardware.
 */
outb_p(0x11, 0x20);      /* ICW1: select 8259A-1 init */
outb_p(0x20 + 0, 0x21); /* ICW2: 8259A-1 IR0-7 mapped to 0x20-0x27 */
outb_p(0x04, 0x21);      /* 8259A-1 (the primary) has a secondary on IRQ2
if (auto_eoi)
        outb_p(0x03, 0x21);
else
        outb_p(0x01, 0x21);

outb_p(0x11, 0xA0);      /* ICW1: select 8259A-2 init */
outb_p(0x20 + 8, 0xA1); /* ICW2: 8259A-2 IR0-7 mapped to 0x28-0x2f */
outb_p(0x02, 0xA1);      /* 8259A-2 is a secondary on primary's IRQ2
outb_p(0x01, 0xA1);
```



port(A=?) info contained in Initialization Control Word

ICW1	0	start init, edge-triggered inputs, cascade mode, 4 ICWs
ICW2	1	high bits of vector #
ICW3	1	primary PIC: bit vector of secondary PIC; secondary PIC: input pin on primary PIC
ICW4	1	ISA=x86, normal/auto EOI