

# 操作系统

# Operating Systems

## L27 键盘

### Keyboard

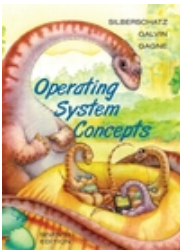
[lizhijun\\_os@hit.edu.cn](mailto:lizhijun_os@hit.edu.cn)

综合楼411室

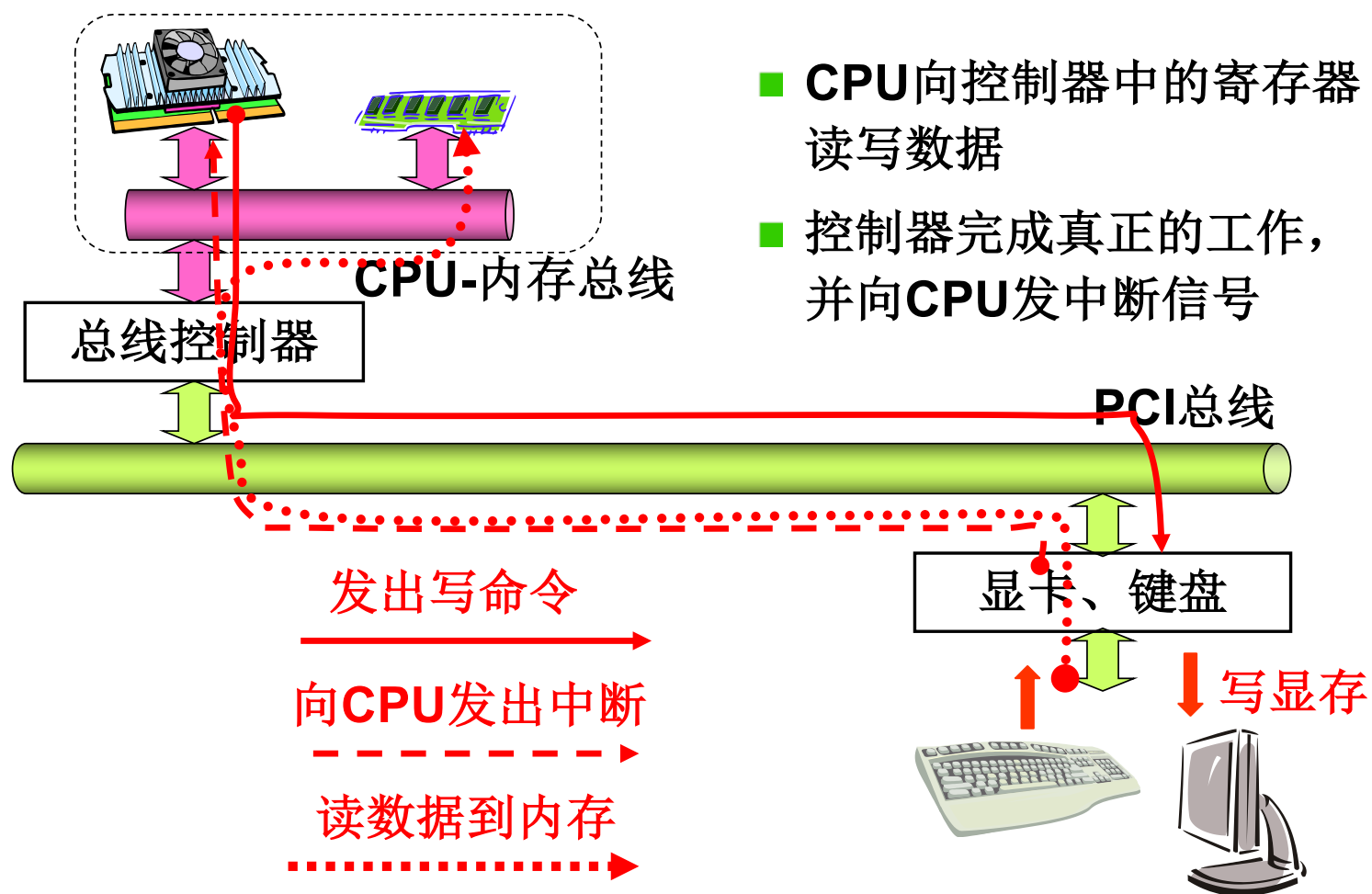
授课教师：李治军

---

**终端设备包括显示器和键盘**



# 让外设工作起来



# 关于键盘的故事从哪里开始？

## ■ 如何使用键盘？

- 对于使用者(人): 敲键盘、看结果
- 对于操作系统: “等着”你敲键盘, 敲了就中断
- 所以故事该从键盘中断开始, 从中断初始化开始...

```
void con_init(void) //应为键盘也是console的一部分
{ set_trap_gate(0x21, &keyboard_interrupt); }
```

在kernel/chr\_drv/keyboard.S中

```
.globl _keyboard_interrupt
_keyboard_interrupt:
    inb $0x60,%al //从端口0x60读扫描码
    call key_table(,%eax,4) //调用key_table+eax*4
    ... push $0    call _do_tty_interrupt
```



# 处理扫描码key\_table+eax\*4

## ■ key\_table是一个函数数组

在kernel/chr\_drv/keyboard.S中

key\_table:

```
.long none,do_self,do_self,do_self //扫描码00-03  
.long do_self, ...,func, scroll, cursor 等等
```

显示字符通常都用此函数处理!

## ■ 扫描码02对应按键1; 01对应ESC; 12对应E等等

mode: .byte 0

do\_self:

lea alt\_map,%ebx

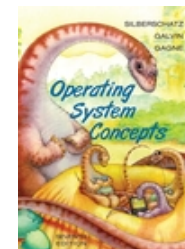
testb \$0x20,mode //alt键是否同时按下 jne 1f

lea shift\_map,%ebx testb \$0x03,mode jne 1f

lea key\_map,%ebx

1:

找到映射表, 如a的  
key\_map映射为a, 而  
shift\_map映射为A

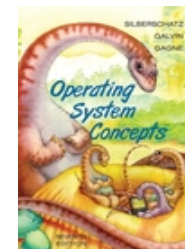


# 从key\_map中取出ASCII码

```
#if defined(KBD_US)
key_map: .byte 0,27 .ascii "1234567890-=" ...
shift_map: .byte 0,27 .ascii "`!@#$%^&*()_+\" ...
#elif defined(KBD_GR) ...
```

■ 继续do\_self, 从1f开始, ebx放的是map起始地址

```
1: movb (%ebx,%eax),%al //扫描码索引, ASCII码→a1
   orb %al,%al   je none //没有对应的ASCII码
   testb $0x4c,mode //看caps是否亮
   je 2f   cmpb $'a',%al   jb 2f
   cmpb $'}',%al   ja 2f   subb $32,%al //变大写
2: testb $??,mode //处理其他模式, 如ctrl同时按下
3: andl $0xff,%eax call put_queue
none: ret
```



## put\_queue将ASCII码放到? con.read\_q

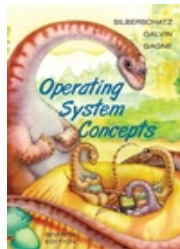
```
put_queue:
    movl _table_list,%edx
    movl head(%edx),%ecx
1:movb %al,buf(%edx,%ecx)
```

```
struct tty_queue *table_list[]={
    &tty_table[0].read_q,
    &tty_table[0].write_q;
    ...};
```

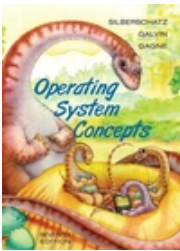
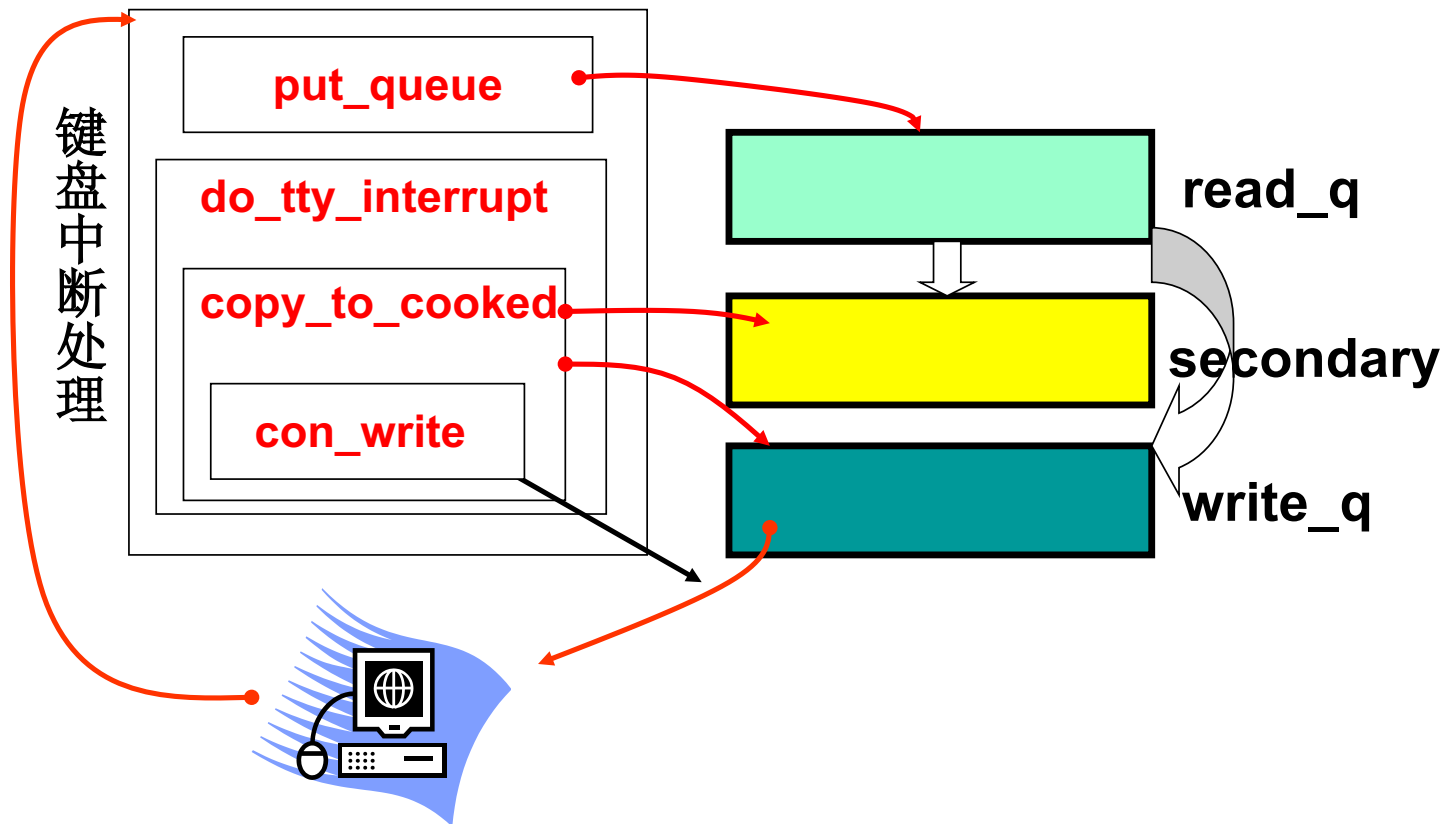
■ 到目前为止还差什么? 对了... “回显”

```
void do_tty_interrupt(int tty) //上面传来的是0
{ copy_to_cooked(tty_table+tty); }
```

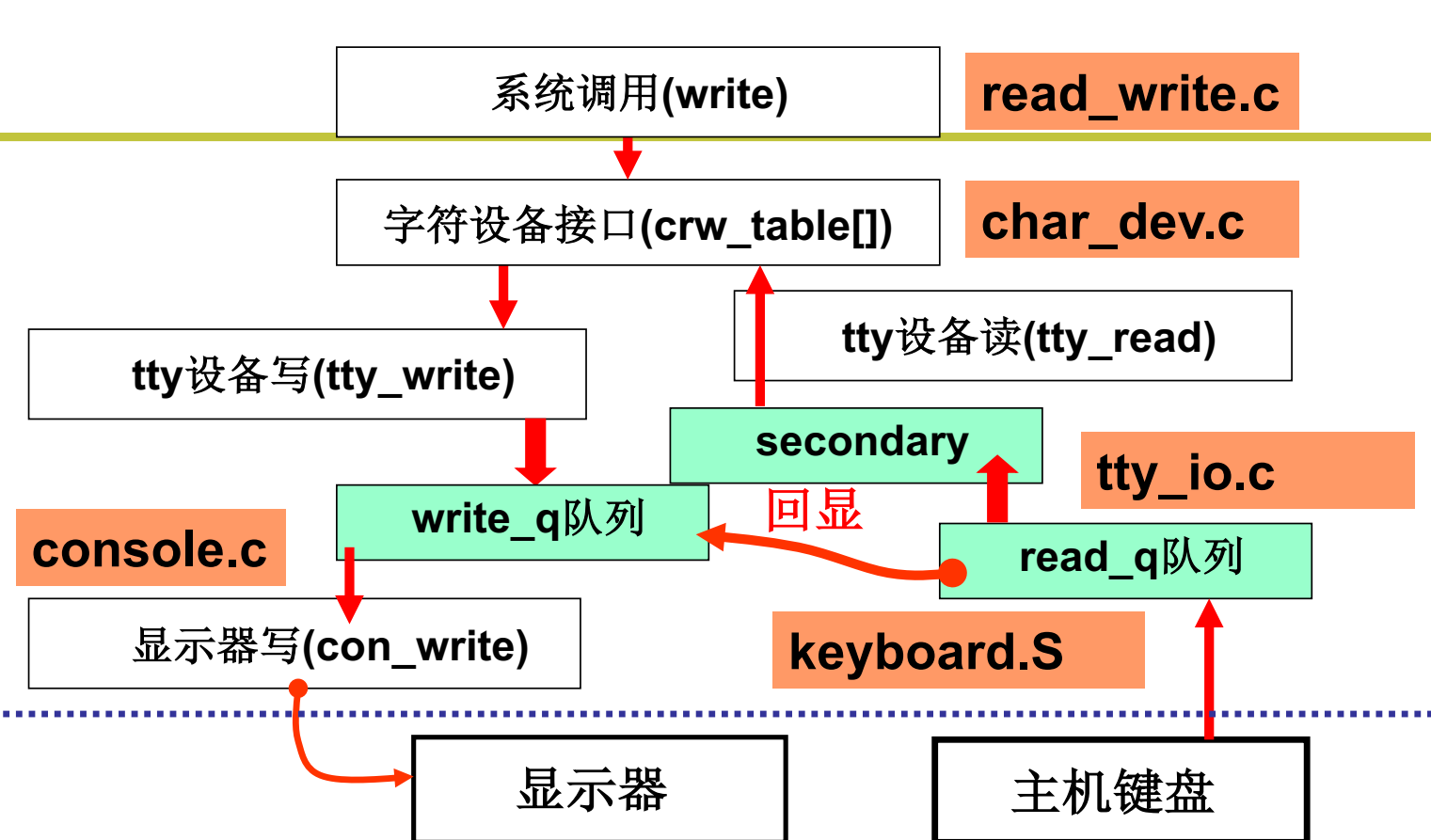
```
void copy_to_cooked(struct tty_struct *tty)
{ GETCH(tty->read_q,c);
  if(L_ECHO(tty)){ //回显,也可以不回显
    PUTC(c,tty->write_q);
    tty->write(tty); } //立刻显示到屏幕上
  PUTC(c,tty->secondary); //完成copy_to_cooked
  ... wake_up(&tty->secondary.proc_list);}
```



# 键盘处理...







问题：如果按下F12让ls的输出为\*，怎么办？如果只让回显为\*，怎么办？如果是让输出的文件为\*，又怎么办？

