

I/O. 对上. everything is file.

对下. 对接设备.

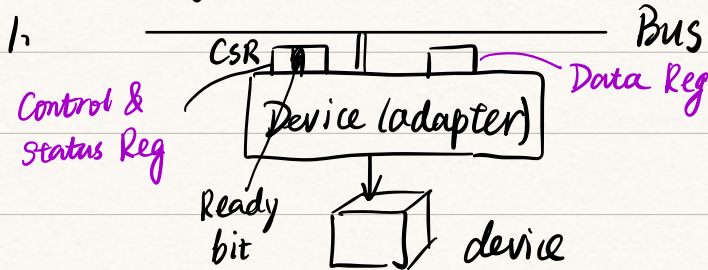
中间 格式转换等

RW: Bottom half

Top-half.

(maybe Interr.) Device

软中断?



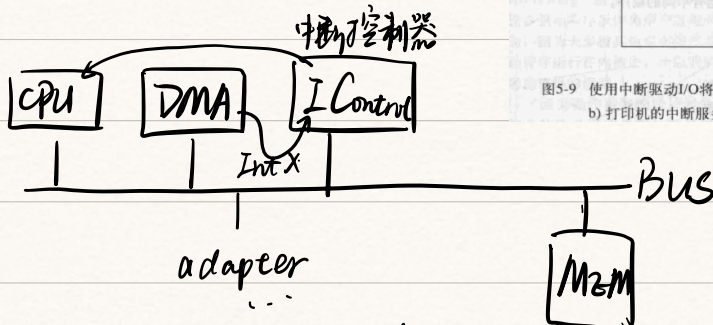
busy-waiting
print a string
while status_reg != Ready;

2. Interrupt driven I/O

这种允许CPU在等待打印机变为就绪的同时做某些其他事情的方式就是使用中断。当打印字符串的系统调用被发出时，如我们前面所介绍的，字符串缓冲区被复制到内核空间，并且一旦打印机准备好接收一个字符时就将第一个字符复制到打印机中。这时，CPU要调用调度程序，并且某个其他进程将运行。请求打印字符串的进程将被阻塞，直到整个字符串打印完。系统调用所做的工作如图5-9a所示。

当打印机将字符打印完并且准备好接收下一个字符时，它将产生一个中断。这一中断将停止当前进程并且保存其状态。然后，打印机中断服务过程将运行。图5-9b所示为打印机中断服务过程的一个粗略的版本。如果没有更多的字符要打印，中断处理程序将采取某个操作将用户进程解除阻塞。否则，它将

3. I/O Using DMA



set-up-DMA-controller().

数据从 MEM → Data-Reg.

scheduler()?

输入/输出

199

输出下一个字符，应答中断，并且返回到中断之前正在运行的进程，该进程将从其停止的地方继续运行。

a)

```
copy_from_user(buffer, p, count);
enable_interrupts();
while (*printer_status_reg != READY);
*printer_data_register = p[0];
scheduler();
```

b)

```
if (count == 0) {
    unblock_user();
} else {
    *printer_data_register = p[i];
    count = count - 1;
    i = i + 1;
}
acknowledge_interrupt();
return_from_interrupt();
```

图5-9 使用中断驱动I/O将一个字符串写到打印机：a) 当打印系统调用被发出时执行的代码；b) 打印机的中断服务过程

不打扰 CPU? DMA 类似于快递员

独立直接写入内存

4. I/O software layers

{ user-level I/O software
Device-independent OS software
device driver
Interrupt handlers
H/W

Recall access disk. superblocks



磁盘本身 dev/hdo.

5. Disk Arm Scheduling Algorithms

最短寻道优先 与 电梯算法

?

6. X window System

x-client x-server



X-server: 从键盘/鼠采集输入、output to screen

↓
X-client: 发送指令

可远程.

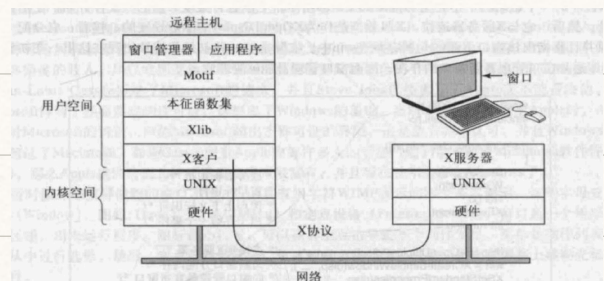


图5-33 MIT X窗口系统中的客户与服务