第十三章 I/O 系统

1. Consider the following I/O scenarios on a single-user PC.
a. A mouse used with a graphical user interface
b. A tape drive on a multitasking operating system (assume no device preallocation is available)
c. A disk drive containing user files

For each of these I/O scenarios, would you design the operating system to use buffering, spooling, caching, or a combination? Would you use polled I/O, or interrupt-driven I/O? Give reasons for your choices.

2. What are the various kinds of performance overheads associated with servicing an interrupt?

1.

a. 缓冲 + 中断驱动 I/O

鼠标是"间歇性产生输入"的交互设备。中断驱动 I/O 能让 CPU 平时处理其他任务，只有鼠标有操作时，才通过"中断"通知 CPU 处理，避免空等

缓冲则用来临时存放鼠标输入的数据 — 因为鼠标产生的速度和 CPU 处理的速度不匹配，缓冲能协调两者。

b. 缓冲 + spooling + 中断驱动 I/O

磁带是比较慢的任务，没有设备预分配，假脱机能把磁带的 I/O 模拟为多任务可共享。缓冲为了填补磁带和 CPU 的速度差；中断驱动 I/O，是因为如果轮询 I/O，浪费 CPU 资源，中断驱动 I/O 更高效

c. 缓冲 + 缓存 + 中断驱动 I/O

缓存可以把频繁访问的文件暂存在内存，下次访问直接内存读，不用读磁盘，提高速度，缓冲用于 I/O 临时存数据。中断驱动 I/O 是因为磁盘操作完成后，发中断通知 CPU，CPU 不用轮询，可以处理其他任务。

2.

① 上下文切换开销

中断发生必须保存当前正在执行程序的上下文，才能转去处理中断；完毕后，恢复上下文，继续执行原来程序

② 中断服务程序的执行开销

中断服务程序的执行本身需要占用CPU。

③ 缓存失效开销

CPU有高速缓存，用来暂存数据。当前执行程序可能有数据存在里面，但中断服务程序的数据可能不存缓存中，CPU需要从内存读，可能增加时间开销

④ 多中断嵌套的开销

当处理中断，又有更高优先级的中断到来，CPU需要再次进行上下文切换，处理完高优先级中断后，再恢复之前的中断上下文处理，会叠加上下文切换开销。