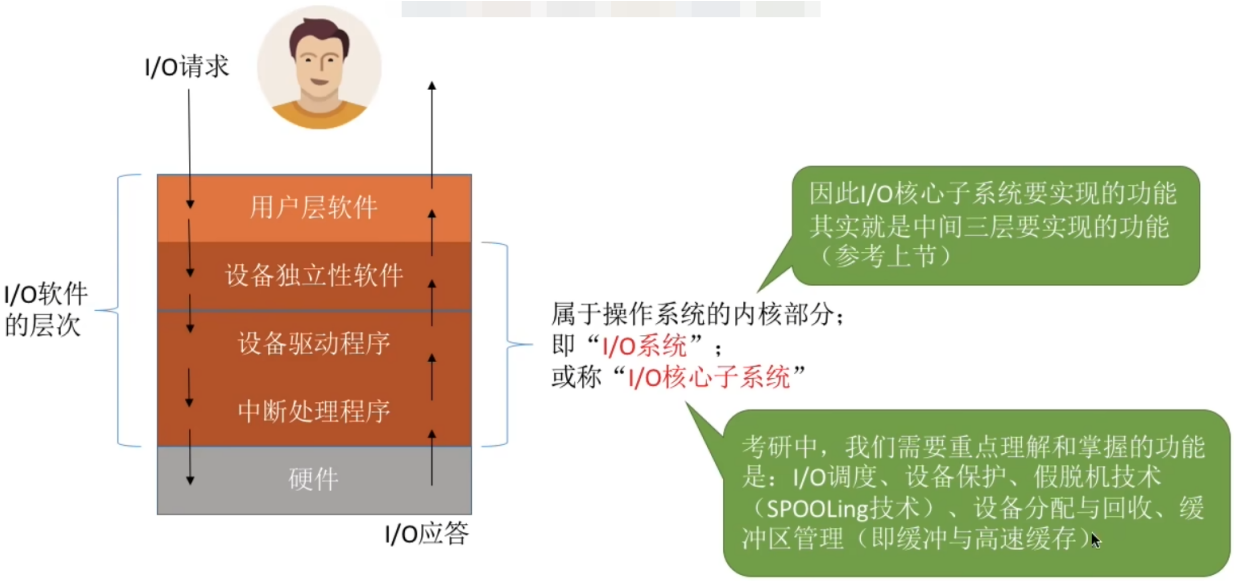
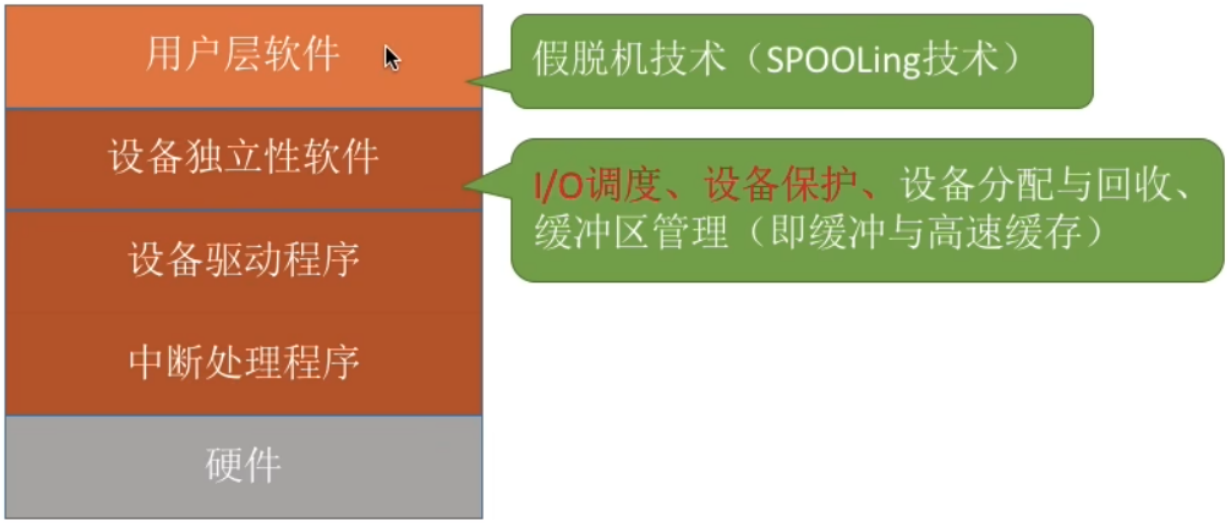
设备独立性软件

I/O核心子系统



这些功能要在哪个层次实现？



I/O调度

用某种算法确定一个好的顺序来处理各个I/O请求。

如：磁盘调度（先来先服务算法、最短寻道优先算法、SCAN算法、C-SCAN算法、LOOK算法、C-LOOK算法）。当多个磁盘I/O请求到来时，用某种调度算法确定满足I/O请求的顺序。

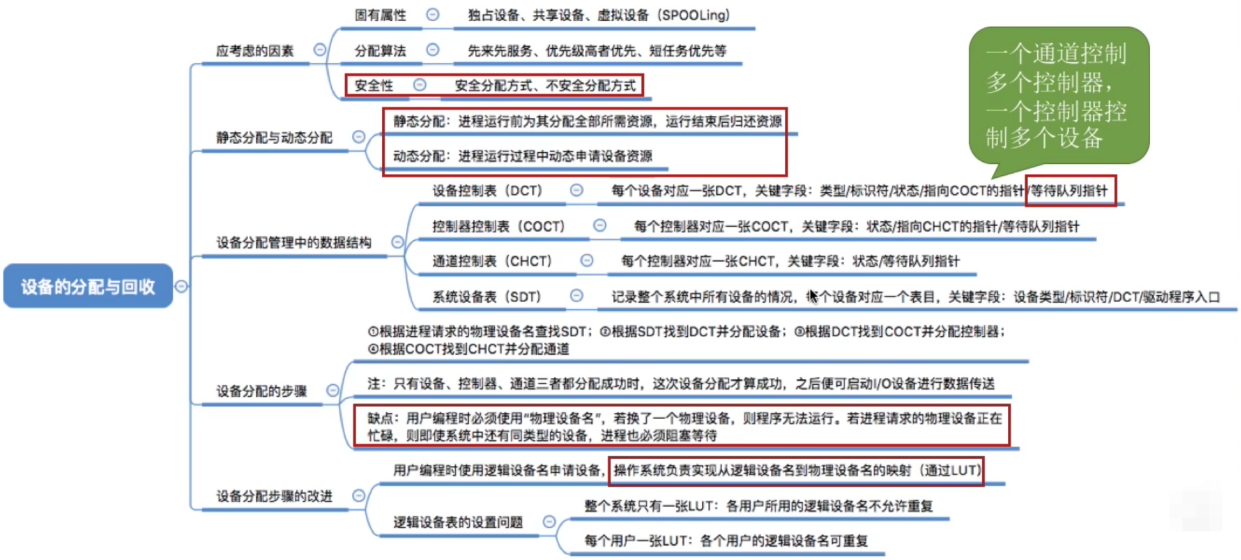
同理，打印机等设备也可以用先来先服务算法、优先级算法、短作业优先等算法来确定I/O调度顺序。

设备保护

操作系统需要实现文件保护功能，不同的用户对各个文件有不同的访问权限（如：只读、读和写等）。

在UNIX系统中，设备被看作是一种特殊的文件，每个设备也会有对应的FCB。当用户请求访问某个设备时，系统根据FCB中记录的信息来判断该用户是否有相应的访问权限，以此实现“设备保护”的功能。（参考文件保护小节）

设备分配与回收



设备分配时应考虑的因素

设备的固有属性可分为三种：独占设备、共享设备、虚拟设备。

- 独占设备

  一个时段只能分配给一个进程（如打印机）

- 共享设备

  可同时分配给多个进程使用（如磁盘），各进程往往是宏观上同时共享使用设备，而微观上交替使用。

- 虚拟设备

  采用SPOOLing技术将独占设备改造成虚拟的共享设备，可同时分配给多个进程使用（如采用SPOOLing技术实现的共享打印机）

设备的分配算法：

- 先来先服务
- 优先级高者优先
- 短任务优先
- ...

从进程运行的安全性上考虑，设备分配有两种方式：

安全分配方式：为进程分配一个设备后就将进程阻塞，本次I/O完成后才将进程唤醒。（eg. 考虑进程请求打印机打印输出的例子）

一个时段内每个进程只能使用一个设备

优点：破坏了请求和保持条件，不会死锁

缺点：对于一个进程来说，CPU和I/O设备只能串行工作

不安全分配方式：进程发出I/O请求后，系统为其分配I/O设备，进程可继续执行，之后还可以发出新的I/O请求。只有某个I/O请求得不到满足时才将进程阻塞。

一个进程可以同时使用多个设备

优点：进程的计算任务和I/O任务可以并行处理，使进程迅速推进

缺点：有可能发生死锁（死锁避免、死锁的检测和解除）

静态分配和动态分配

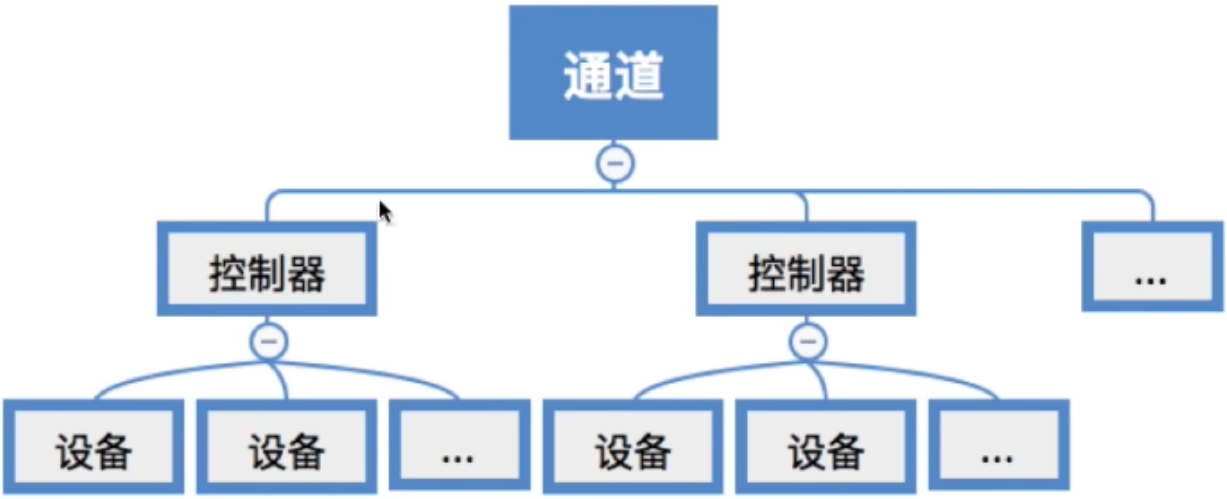
静态分配：进程运行前为其分配全部所需资源，运行结束后归还资源

破坏了请求和保持条件，不会发生死锁

动态分配：进程运行过程中动态申请设备资源

设备分配管理中的数据结构

“设备、控制器、通道”之间的关系：



一个通道可控制多个设备控制器，每个设备控制器可控制多个设备

设备控制表（DCT）：系统为每个设备配置一张DCT，用于记录设备情况

设备控制表（DCT）	
设备类型	如：打印机/扫描仪/键盘
设备标识符	即物理设备名，系统中的每个设备的物理设备名唯一
设备状态	忙碌/空闲/故障...
指向控制器表的指针	每个设备由一个控制器控制，该指针可找到相应控制器的信息
重复执行次数或时间	当重复执行多次I/O操作后仍不成功，才认为此次I/O失败
设备队列的队首指针	指向正在等待该设备的进程队列（由进程PCB组成队列）

注："进程管理"章节中曾经提到过"系统会根据阻塞原因不同，将进程PCB挂到不同的阻塞队列中"

控制器控制表（COCT）：每个设备控制器都会对应一张COCT。操作系统根据COCT的信息对控制器进行操作和管理。

控制器控制表（COCT）	
控制器标识符	各个控制器的唯一ID
控制器状态	忙碌/空闲/故障...
指向通道表的指针	每个控制器由一个通道控制，该指针可找到相应通道的信息
控制器队列的队首指针	
控制器队列的队尾指针	指向正在等待该控制器的进程队列（由进程PCB组成队列）

通道控制表（CHCT）：每个通道对应一张CHCT。操作系统根据CHCT的信息对通道进行操作和管理。

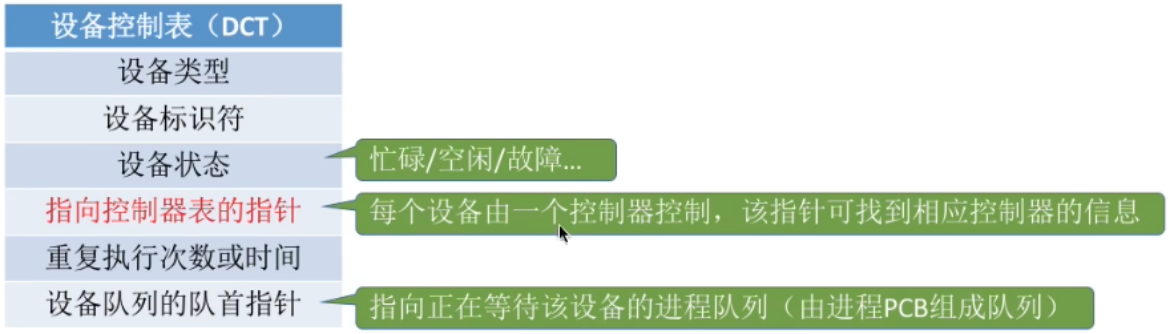
通道控制表（CHCT）	
通道标识符	各个通道的唯一ID
通道状态	忙碌/空闲/故障...
与通道连接的控制器表首址	可通过该指针找到该通道管理的所有控制器相关信息（COCT）
通道队列的队首指针	
通道队列的队尾指针	指向正在等待该通道的进程队列（由进程PCB组成队列）

系统设备表（SDT）：记录了系统中全部设备的情况，每个设备对应一个表目。

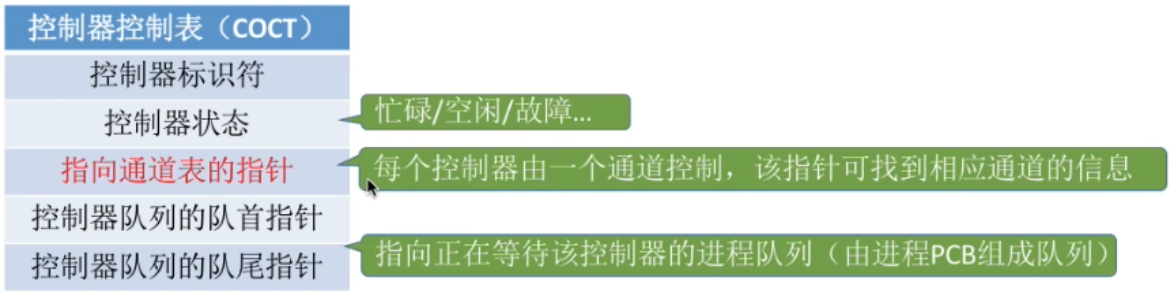
设备分配的步骤

1. 根据进程请求的物理设备名查找SDT（注：物理设备名是进程请求分配设备时提供的参数）

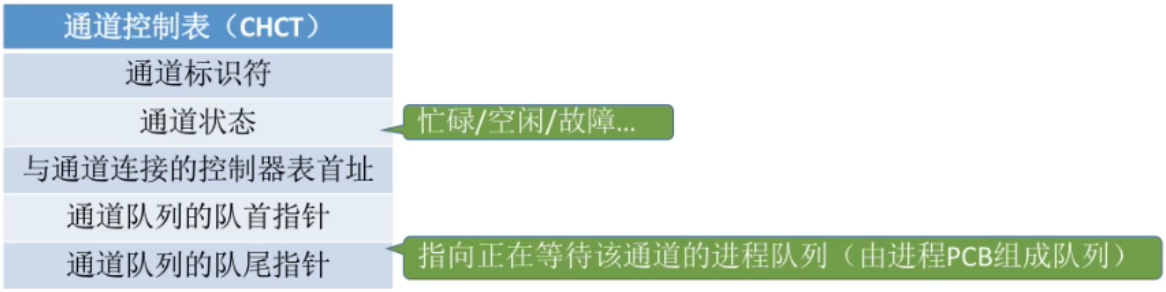
2. 根据SDT找到DCT，若设备忙碌则将进程PCB挂到设备等待队列中，不忙碌则将设备分配给进程。



3. 根据DCT找到COCT，如控制器忙碌则将进程PCB挂到控制器等待队列中，不忙碌则将控制器分配给进程。

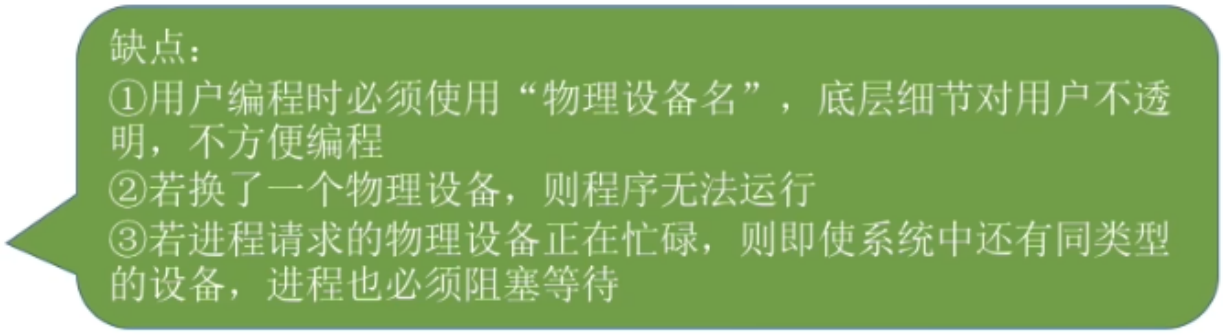


4. 根据COCT找到CHCT，若通道忙碌则将进程PCB挂到通道等待队列中，不忙碌则将通道分配给进程。



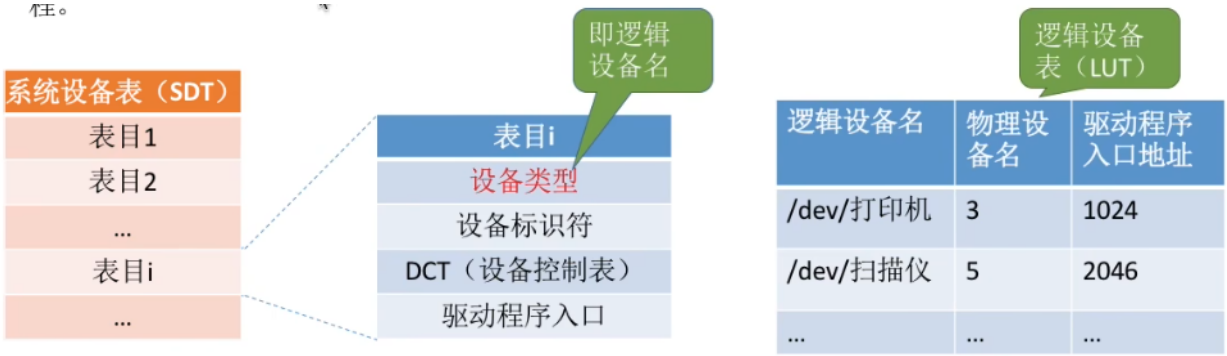
注：只有设备、控制器、通道三者都分配成功时，这次设备分配才算成功，之后便可启动I/O设备进行数据传送

设备分配步骤的改进



改进方法：建立逻辑设备名与物理设备名的映射机制，用户编程时只需要提供逻辑设备名

1. 根据进程请求的逻辑设备名查找SDT（注：用户编程时提供的逻辑设备名其实就是设备类型）
2. 查找SDT，找到用户进程指定类型的、并且空闲的设备，将其分配给该进程。操作系统在逻辑设备表（LUT）新增一个表项。
3. 根据DCT找到COCT，如控制器忙碌则将进程PCB挂到控制器等待队列中，不忙碌则将控制器分配给进程。
4. 根据COCT找到CHCT，若通道忙碌则将进程PCB挂到通道等待队列中，不忙碌则将通道分配给进程。



逻辑设备表（LUT）

逻辑设备名	物理设备名	驱动程序入口地址
/dev/printer	3	1024
/dev/tty	5	2046
…	…	…

逻辑设备表（LUT）建立了逻辑设备名与物理设备名之间的映射关系。

某用户进程第一次使用设备时使用逻辑设备名向操作系统发出请求，操作系统根据用户进程指定的设备类型（逻辑设备名）查找系统设备表，找到一个空闲设备分配给进程，并在LUT中增加相应表项。

如果之后用户进程再次通过相同的逻辑设备名请求使用设备，则操作系统通过LUT表即可知道用户进程实际要使用的是哪个物理设备了，并且也能知道该设备的驱动程序入口地址。

逻辑设备表的设置问题：

整个系统只有一张LUT：各用户所用的逻辑设备名不允许重复，适用于单用户操作系统

每个用户一张LUT：不同用户的逻辑设备名可重复，适用于多用户操作系统