

I/O软件设计

分层的设计思想

- 把I/O软件组织成多个层次
- 每一层都执行操作系统所需要的功能的一个相关子集，它依赖于更低一层所执行的更原始的功能，从而可以隐藏这些功能的细节；同时，它又给高一层提供服务
- 较低层考虑硬件的特性，并向较高层软件提供接口
- 较高层不依赖于硬件，并向用户提供一个友好的、清晰的、简单的、功能更强的接口

清晰的、简单的、功能更强的一个接口 那这就是 I/O



把设备当作一个逻辑资源来处理，不关心实际控制设备的细节



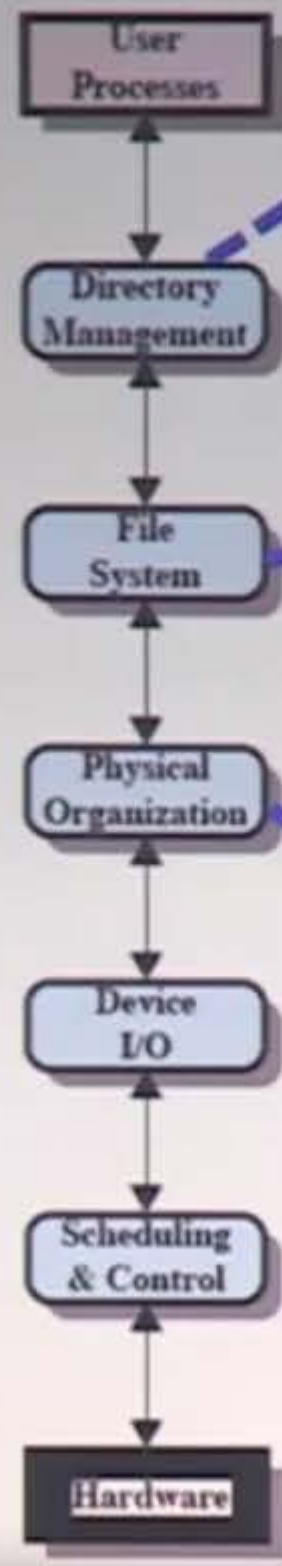
外部设备

请求的操作和数据被转换成适当的I/O指令序列、通道命令和控制器指令

I/O操作排队、调度；处理中断，收集并报告I/O状态



通信接口



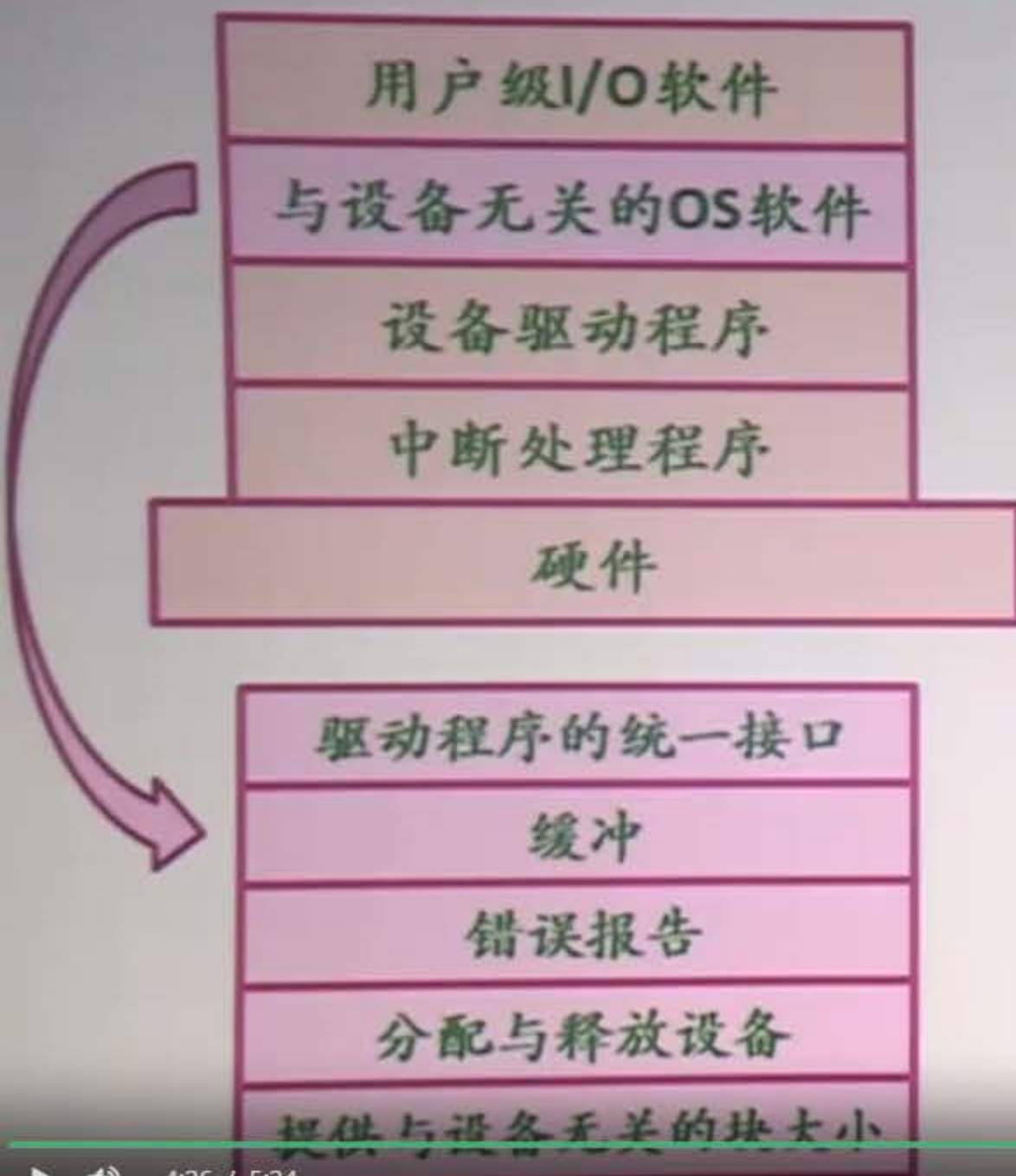
文件系统

符号文件名→标识符→文件描述符表或索引表→访问文件

处理文件的逻辑结构，以及用户指定的打开、关闭、读、写等操作，还管理访问权限

考虑磁盘的物理磁道和扇区结构，逻辑访问必须转换成物理磁盘地址；磁盘存储空间和内存缓冲区的分配

I/O软件层次



- (1) 用户进程层执行输入输出系统调用，对I/O数据进行格式化，为假脱机输入/输出作准备
- (2) 独立于设备的软件实现设备的命名、设备的保护、成块处理、缓冲技术和设备分配
- (3) 设备驱动程序设置设备寄存器、检查设备的执行状态
- (4) 中断处理程序负责I/O完成时，唤醒设备驱动程序进程，进行中断处理
- (5) 硬件层实现物理I/O的操作



设备独立性（设备无关性）

用户编写的程序可以访问任意I/O设备，无需事先指定设备

好处：

设备分配时的灵活性
易于实现I/O重定向

从用户角度：用户在编制程序时，使用逻辑设备名由系统实现从逻辑设备到物理设备（实际设备）的转换，并实施I/O操作

从系统角度：设计并实现I/O软件时，除了直接与设备打交道的低层软件之外，其他部分的软件不依赖于硬件



下面我们介绍一个 I/O 软件的组成 I/O 软件的设计呢，实际上就是一个分层的设计思想 我们把 I/O 软件组织成多个层次 每一层呢都执行操作系统，所需要的功能的一个相关的子集 那么这一层往往依赖于更低一层所执行的更原始的功能 通过这种分层呢，可以隐藏底层的功能细节 并且呢给高层呢提供相应的服务 所以较低层呢，更多的是考虑硬件的特性 而较高层呢，是考虑一些通用的功能 高层不依赖于硬件 这样就给用户提供一个友好的、清晰的、简单的、功能更强的一个接口 那这就是 I/O 软件设计的分层的设计思想 下面我们针对不同类型的 I/O 设备 来介绍一下，它们的 I/O 软件的分层设计思想 首先我们先看一下是普通的一些外部设备 那么然后呢，这是一个 网卡，就是通信相关的接口 另外呢，就是典型的文件系统 对于普通的外部设备呢，我们可以看到它分成了这么几层 第一层呢是逻辑 I/O 层。它把设备 呢当成是一个逻辑的资源来进行处理 并不关心底层的设备的细节 而跟细节相关的呢，是在驱动 I/O 这一层来完成的 那么它会把请求的操作和数据 被转换成适当的 I/O 控制指令 通过这个指令序列来对设备进行相应的控制 之后呢，我们会看到一个调度和控制层 它的主要工作呢，是把各种各样的 I/O 请求呢进行排队 啊，进行相应的调度，以达到一个合理的性能考虑 那么我们来看看文件系统 那么文件系统的话，我们知道它有一个目录管理 它是把符号文件名通过标识 把它转换成文件的描述符，或者是索引表来访问文件 然后这一层呢，是文件系统的一些逻辑结构，然后用户做的一些 打开呀、关闭呀、这个读写这些操作，还有一些访问管理权限的这样一些考虑，在文件系统这一层 然后就是物理磁盘的组织，啊，物理磁盘的组织呢 包括磁道呀、扇区的结构，还有逻辑访问必须转换成一个 物理磁盘地址这样一个过程。那么这就是 针对不同类型的设备，那么我们的 I/O 软件的层次呢是不一样的。那么我们的 I/O 软件层次呢，通常分成了四层 第一层呢是用户级 I/O 软件。所谓用户级 I/O 软件 指的是用户进程在执行输入输出的系统调用 或者是对 I/O 数据进行格式化，或者是为 假脱机，也就是 SPOOLing 的输入输出做准备。那么这呢都属于这一层 所应该完成的工作。第二层呢，是与设备无关的软件层 那么它呢是独立于设备的一些相关工作，比如说设备的命名 设备的保护，还有成块的处理，以及缓冲技术和设备分配都属于 设备无关的 I/O 软件层。那么第三层呢，是设备驱动 程序，主要是设置设备的寄存器，检查设备的执行状态 第四层呢，是中断处理程序，主要是完成 I/O 结束后的一些处理工作 那么与设备无关的软件又可以分为 这样几个层次，比如说驱动程序的统一接口 包括缓冲技术、对错误的报告 对设备的分配与回收，以及提供与设备无关的

尺寸、大小。这都是设备无关层的一些具体工作 那么什么是设备无关性和设备独立性这个概念呢？ 那么它指的是，用户编写程序的时候可以访问任意的 I/O 设备，无需指定相应的设备 那么从用户的角度来看呢，就是指的用户在编制程序的时候，用的是逻辑设备名 由操作系统来完成将逻辑设备到物理设备的一个转换过程，并且实施 I/O 操作 那么从系统的角度呢，它是设计并实现 I/O 软件的时候呢，那么除了直接与设备打交道的底层软件之外，而其它部分和 硬件无关。啊，这都是指的是设备独立性这个概念的一个 特性。 有了设备独立性，它的好处就是说，在设备分配的时候非常灵活，并且容易实现 I/O 重定向