

- 1

- 程序控制I/O: I/O操作由程序发起, 并等待操作完成。数据的每次读写通过CPU
- 中断驱动方式: I/O操作由程序发起, 在操作完成时, 由外设向CPU发出中断, 通知该程序。数据的每次读写通过CPU
- 直接存储访问技术: 由程序设置DMA控制器的若干寄存器值, 然后发起I/O操作; DMA控制器完成内存与外设的成批数据交换; 在操作完成时由DMA控制器向CPU发出中断
- I/O通道控制技术: I/O通道是专门负责输入输出的处理器, 独立于CPU, 有自己的指令体系。可执行由通道指令组成的通道程序, 因此可以进行较为复杂的I/O控制。通道程序通常由操作系统所构造, 放在内存里。

- 2

- 设备独立性
 - 用户程序的设备独立性: 用户程序使用逻辑设备名, 系统实际执行时, 映射到物理设备名
 - I/O软件的设备的独立性: 除了直接与设备打交道的底层软件外, 其余部分软件不依赖于设备, 可提高设备管理软件效率
 - 为实现设备独立性, 系统应为每个用户进程配置一张用于联系逻辑设备名和物理设备名的映射表, 表中一般应包含逻辑设备名、物理设备名和驱动程序入口地址。

- 3

- 缓冲技术可提高外设利用率, 匹配CPU与外设的不同处理速度; 减少对CPU的中断次数; 提高CPU和I/O设备之间的并行性
- 利用单缓冲技术所用的时间是 $(100\mu s + 50\mu s) * 10 + 50\mu s = 1550\mu s$
- 利用双缓冲技术所用的时间是 $100\mu s * 10 + 100\mu s = 1100\mu s$

- 4

- 先来先服务: $(10 - 0) * 6ms + (35 - 10) * 6ms + (35 - 20) * 6ms + (70 - 20) * 6ms + (70 - 2) * 6ms + (3 - 2) * 6ms + (38 - 3) * 6ms = 1224ms$
- 最短寻道时间优先 $((2 - 0) + (3 - 2) + (10 - 3) + (20 - 10) + (35 - 20) + (38 - 35) + (70 - 38)) * 6ms = 420ms$
- Look: $((20 - 15) + (35 - 20) + (38 - 35) + (70 - 38) + (70 - 10) + (10 - 3) + (3 - 2)) * 6ms = 738ms$