

**3、什么是存储器映射 I/O？为什么在一些情况下要使用它？**

答：内存映射的 I/O 将 I/O 寄存器放在正常的内存空间中，因此它们可以像其他内存位置一样被访问。存储器映射 I/O，将设备数据缓冲区按内存地址空间进行统一编址，I/O 寄存器是内存地址空间的一部分，被分配唯一的一个内存地址，位于地址空间的顶端。这样可以在不使用基本 I/O 操作函数 read 和 write 的情况下执行 I/O 操作。

**4、尝试解释什么是 DMA。为什么要使用它？**

答：DMA，“Direct Memory Access”，直接存储器访问。大多数输入/输出包括在输入/输出设备和内存中连续位置之间重复传输字节。将数据从一个地址空间复制到另外一个地址空间，CPU 只需要干预 I/O 操作的开始和结束，而传输动作本身由 DMA 控制器完成，CPU 可以去做其他工作，提高 CPU 的效率。适用于高速设备。

**11、以下的工作各在 4 个 I/O 软件层的哪一层完成？**

(a)为一个磁盘读操作计算磁道、扇区、磁头。

答：设备驱动程序

(b)维护一个最近使用的块的缓冲。

答：设备无关软件

(c)向设备寄存器写命令。

答：设备驱动程序

(d)检查用户是否有权使用设备。

答：设备无关软件

(e)将二进制整数转换成 ASCII 码以便迅速打印。

答：用户进程

**12、为什么打印的输出文件在打印前通常都假脱机输出在磁盘上？**

答：如果一个用户进程占用打开代表打印机的字符设备文件，但却长时间不使用，那么其他进程也不能打印任何东西。因此，将需要打印的文件假脱机输出在磁盘上，由守护进程打印文件，来防止用户进程直接使用打印机，可以解决某些进程不必要地长期空占打印机的问题。

**14、考虑图 3.10。假设在步骤(o)中 C 需要 S 而不是 R，这将会导致死锁吗？如果既需要 S 又需要 R 呢？**

答：如果 C 需要 S 而不是 R，这不会导致死锁。因为进程 A 已经执行完成，资源 S 可以被释放分配给 C。

如果 C 既需要 S 也需要 R 仍不会导致死锁。因为进程 A 已经执行完成，资源 S 和 R 都可以释放分配给 C，C 执行完成后，再分配给进程 B。

**15、仔细观察图 3.13(b)。如果 D 再多请求一个单位，将会导致一个安全的状态还是不安全的状态？如果请求来自 C 而不是 D 呢？**

答：D 再多请求一个单位，状态仍然是安全的。可用的两个单位可以先让 C 完成，然后释放出其占用的资源，再让 D、B、A 依次完成。

C 再多请求一个单位，状态不安全。此时可用资源只有两个单位，此时 A、B、C、D 四个都不能满足要求。

**17、假设图 3.15 中的进程 A 请求最后一台磁带驱动器。这会导致死锁吗？**

答：不会导致死锁；若立即满足 A 的请求，则可用资源向量减为  $A=(0020)$ 。再将每个进程所需要的资源与可用资源向量作比较，进程 D 可以执行，执行结束后释放出它所占有的资源，进程 A、B、C、E 也可以依次执行结束。因此，所得到的系统状态仍然是安全的，不会导致死锁。

**22、银行家算法在一个有  $m$  个资源类和  $n$  个进程的系统中运行。在  $m$  和  $n$  都很大的情况下，为检查状态是否稳定而进行的操作次数正比于  $m^a n^b$ 。a 和 b 的值为多少？**

答：多种资源的银行家算法维护了两个矩阵分别记录每个进程每种资源的已分配数量和仍需要的数量。找出可执行的进程需要将每个进程仍需要的资源数量（对应矩阵的一行）与可用资源向量进行比较，需要比较  $m$  次，一共  $n$  个进程，则需要  $mn$  次操作找出可以执行的进程。因为一共有  $n$  个进程，故需要  $mn \times n = m^1 n^2$  次操作。

所以  $a = 1, b = 2$ 。

**28、磁盘请求以 10, 22, 20, 2, 40, 6, 38 柱面的次序到达磁盘驱动器。寻道时每个柱面移动需要 6ms，计算以下寻道时间：（所有情况下磁头臂起始都位于柱面 20。）**

(a)先到先服务。

答：选择请求的顺序为：10, 22, 20, 2, 40, 6, 38

磁盘臂移动的柱面数依次为：10, 12, 2, 18, 38, 34, 32

寻道时间： $(10+12+2+18+38+34+32) \times 6\text{ms} = 876\text{ms}$

(b)下一个最邻近柱面。

答：选择请求的顺序为：20, 22, 10, 6, 2, 38, 40

寻道时间： $(0+2+12+4+4+36+2) \times 6\text{ms} = 360\text{ms}$

(c)电梯算法（起始移动向上）。

答：选择请求的顺序为：20, 22, 38, 40, 10, 6, 2

寻道时间： $(0+2+16+2+30+4+4) \times 6\text{ms} = 348\text{ms}$