

8.2 磁盘调度算法

操作系统的任务之一就是有效地使用硬件。对磁盘驱动器来说，满足这一要求意味着要有较快的访问速度和较宽的磁盘带宽。访问时间包括两个主要部分是寻道时间和旋转延迟。寻道时间是磁臂将磁头移动到包含目标扇区的柱面的时间。旋转延迟是磁盘需要将目标扇区转动到磁头下的时间。磁盘带宽是所传递的总的字节数除以从服务请求开始到最后传递结束时的总时间。可以通过使用适当的访问顺序来调度磁盘I/O请求，提高访问速度和带宽。

每当一个进程需要对磁盘进行I/O操作，它就向操作系统发出一个系统调用。该调用请求指定了一些信息：

- 操作是输入还是输出？
- 所传输的磁盘地址是什么？
- 所传输的内存地址是什么？
- 所传输的扇区数是多少？

如果所需的磁盘驱动器和控制器空闲，那么该请求会马上处理。如果磁盘驱动器或控制器忙，那么任何新的服务请求都会加到该磁盘驱动器的待处理请求队列上。对于一个有多个进程的多道程序设计系统，磁盘队列可能有多个待处理请求。因此，当完成一个请求时，操作系统可以选择处理哪个待处理请求。那么操作系统该如何选择呢？有多个磁盘调度算法可供使用，接下来将会加以讨论

磁盘调度算法用以改善磁盘的平均寻道时间。典型算法有FCFS算法、最短寻道时间优先算法SSTF、SCAN算法、C-SCAN算法。

目前常用的磁盘调度算法有：先来先服务、最短寻道时间优先、扫描算法和循环扫描算法等。

(1) 先来先服务算法FCFS (First Come First Served)

这是一种最简单的磁盘调度算法。它根据进程请求访问磁盘的先后次序进行调度。此算法的优点是公平、简单，且每个进程的请求都能依次得到处理，不会出现某一进程的请求长期得不到满足的情况。但此算法由于未对寻道进行优化，致使平均寻道时间可能较长。

假定有一个磁盘请求队列，其I/O对各柱面上块的请求序列如下：（磁道编号从0 - 199）

98, 183, 37, 122, 14, 124, 65, 67

如果当前磁头位置位于53，图8.4显示了FCFS调度算法的磁头移动路径，总的磁头移动为640柱面。

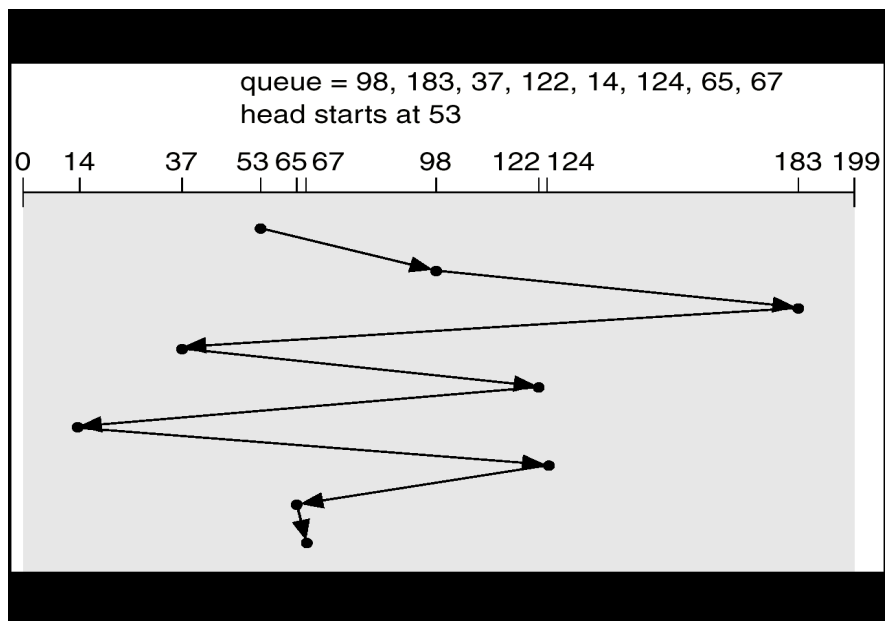


图8.4 FCFS磁盘调度

(2) 最短寻道时间优先算法SSTF (Shortest Seek Time First)

该算法总是为那些与当前磁头所在的磁道距离最近请求服务。也就是执行寻道时间最短的那个I/O请求。这种调度算法有较好的平均寻道时间。SSTF较之 FCFS有较好的寻道性能，故曾被广泛采用。

对于上面的磁盘请求队列的例子，如果当前磁头位置位于53。如图8.5显示了SSTF调度算法的磁头移动路径，磁头移动的总距离是236柱面。

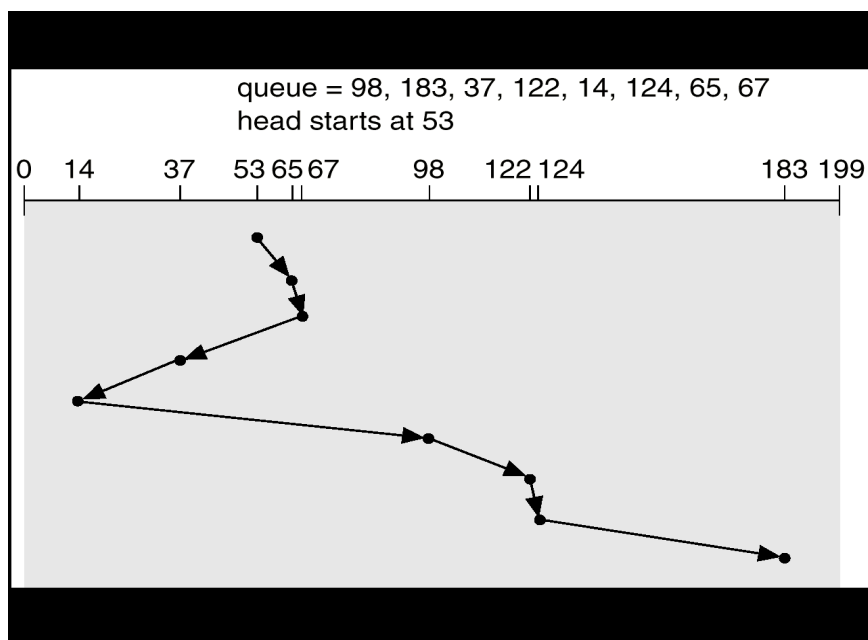


图8.5 SSTF磁盘调度

(3) 扫描算法 (SCAN)

SSTF算法虽然获得较好的寻道性能，但它可能导致某些进程长时间的得不到服务（称之为饥饿现象）。因为只要不断有新进程到达，且其所要访问的磁道与磁头当前所在磁道的距离较近，这种新进程的I/O请求必被优先满足。对 SSTF算法略加修改后所形成了SCAN算法。

该算法不仅考虑到欲访问的磁道与当前磁道的距离，更优先考虑的是磁头的当前移动方向。即当磁头正在自里向外运动时，SCAN算法要选择的下一个访问对象是其欲访问的磁道在当前磁道之外，又是距离最近的。直至再无更外的磁道需要访问时，才将磁臂换向，自外向里运动。从而避免了饥饿现象的出现。由于这种算法中磁头移动的规律象电梯的运行，所以又称为电梯调度算法。

对于上面的磁盘请求队列的例子，如果当前磁头位置位于53，且磁头向0方向移动。如图8.6显示了SCAN调度算法的磁头移动路径，磁头移动的总距离是208柱面。

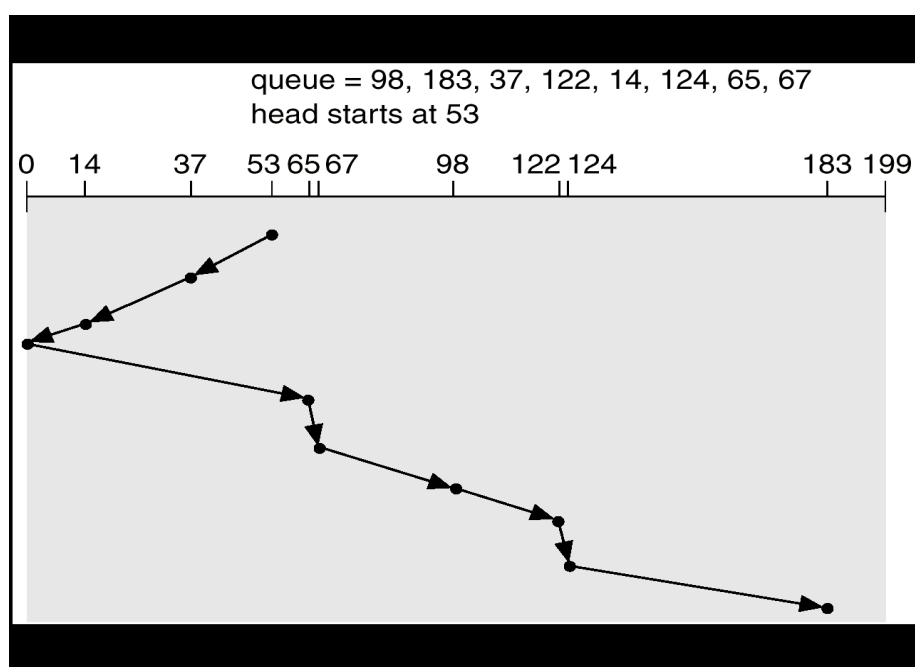


图8.6 SCAN磁盘调度

(4) 循环扫描算法C-SCAN (Circular SCAN)

这是SCAN算法的一种变种算法，是为了提供更均匀的等待时间而设计的。C-SCAN算法规定磁头只能单向运动（自里向外），当磁头运动到最外面的被访问磁道时，磁头立即返回到最里面的欲访的磁道，即将最小磁道号紧接着最大磁道号构成循环，进行扫描。

对于上面的磁盘请求队列的例子，如果当前磁头位置位于53，且磁头向199方向移动。如图12.7显示了C-SCAN调度算法的磁头移动路径，磁头移动的总距离是322柱面。

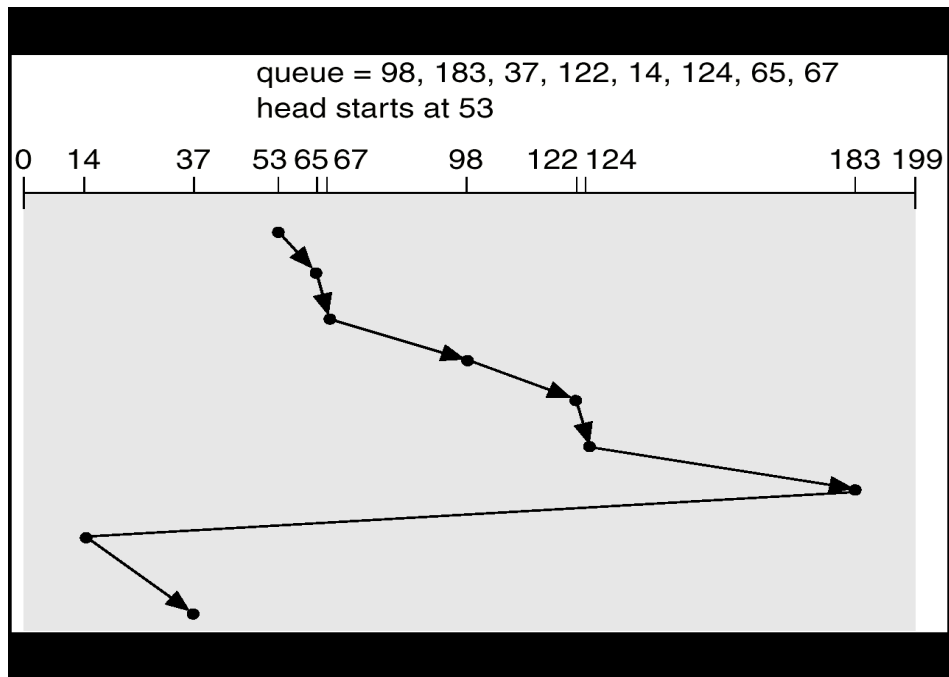


图8.7 C-SCAN磁盘调度