OS HW Chapter 11

3170105860 彭子帆

**11.11**

1. FCFS是按照顺序进行“先来先服务”的思想，因此不会饥饿。而其他所有调度方式中总会使一些请求等待时间过久才能响应，因此会“饥饿”很多请求。SCAN中如果在磁头掉头后立刻有一个附近的请求出现，就有可能导致该饥饿，C-SCAN中同理也是如此。LOOK和C-LOOK是掉头后如果出现比当前最大（最小）的还要大（小）的请求那么也会对其造成饥饿。
2. SCAN中可以为一个请求设置一个年龄属性或者优先级属性，年龄或者优先级随着时间增加增大，如果一个请求优先级或年龄大于所有其他的，那么就优先对其进行处理，从此处接着向SCAN此时磁头所朝向方向进行扫描。
3. 因为一个多用户系统中，多个用户同时发送请求，如果一直处理其中一个或多个用户，会造成许多用户饥饿，造成长时间的延迟，我们要保证系统不能有很多次过长时间的响应时间。
4. 很多内核的I/O调度需要提前进行，比如写文件系统的元数据。页面调度以及页面置换都需要有更高的优先级去

**11.13**

1. **FCFS:**

如果是FCFS那么按顺序，调度序列是：2150，2069，1212，2296，2800，544，1618，356，1523，4965，3681

经过计算那么计算总的调度序列的磁头移动距离为：

81+857+1084+504+2256+174+1262+1167+3442+1284 = 12111

1. **SCAN:**

如果是SCAN，那么按照当前的磁针的方向从1805->2150，那么可以推算朝正方向运行中。因此序列可以得出：2150，2296，2800，3681，4965，4999，转弯儿，2069，1681，1523，1212，544，356

那么计算总的调度序列的磁头移动距离为：

146+504+881+1284+34+2930+388+158+311+668+188 = 7492

1. **C-SCAN:**

如果是C-SCAN，与SCAN类似，只需要当其到达4999时直接归0，再继续扫描即可，因此序列如下：2150，2296，2800，3681，4965，4999，0，356，544，1212，1523，1681，2069

那么总的磁头移动距离长度为：

146+504+881+1284+34+4999+356+188+668+311+158+388 = 9917

**11.14**

1. 根据经典物理学，一个有着恒定加速度的物体移动时间t后的距离长度为v0t+1/2 at2，我们知道v0=0，所以d = 1/2 at2，根据题目的描述我们可以容易地注意到总距离长度为1/2aT12+1/2aT22, 其中 T1 、T2 分别为第一个半段和第二个半段的时间长度，并且他们相等，所以d = aT12, and ，其中T为总时间。
2. 根据这道题，我们可以轻松的发现两个等式如下：

1 = x + y, 18 = x + y

所以我们能计算出

y = 0.2438898, x = 0.7561102

所以该等式为：t = 0.7561102 + 0.2438898

1. 根据11.13我们计算出的结果和上一问的结果我们可以容易地计算出：

**FSFC:** T = 10 \* (0.7561102) + 0.2438898 \* (）= 85.04

**SCAN:** T = 10 \* (0.7561102) + 0.2438898 \*（）= 68.4

**C-SCAN:** T = 10 \* (0.7561102) + 0.2438898 \* （） = 77.05

因此可以得出SCAN最快。

d. 通过上一问得出最快是SCAN，其相比FCFS节省16.64，因此其加速百分比为：16.64/85.04 \* 100% = 19.58%，加速了19.58%。

**11.15**

1. 7200rpm即7200 rounds per minutes, 即120转每秒，0.12转每毫秒，因此1/0.12 = 8.33毫秒每转，因此如果磁头转动的每种长度概率相同的时候，那么大概平均半转，则大概平均延迟为4.167毫秒，因此平均延迟为4.167毫秒。
2. 根据上题算出的公式：t = 0.7561102 + 0.2438898，当t为4.167毫秒的时候，长度为195.59个cylinders，所以我们能够在一次平均转动延迟中找到寻找195个tracks。