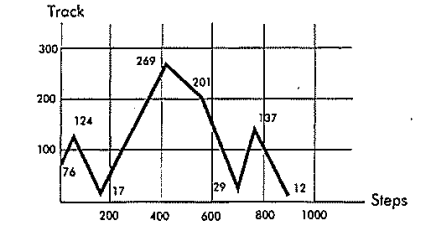
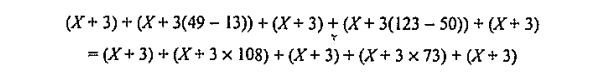
rotating disk

怎么计算用时

X是启动用时，3是一个track所用时间，123-12是track差

FCFS随着顺序来的， first come first served

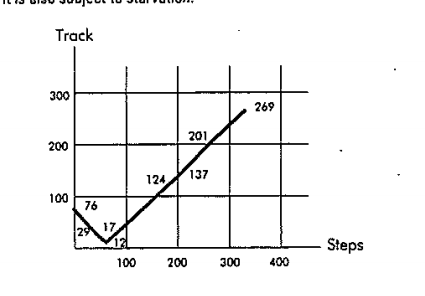




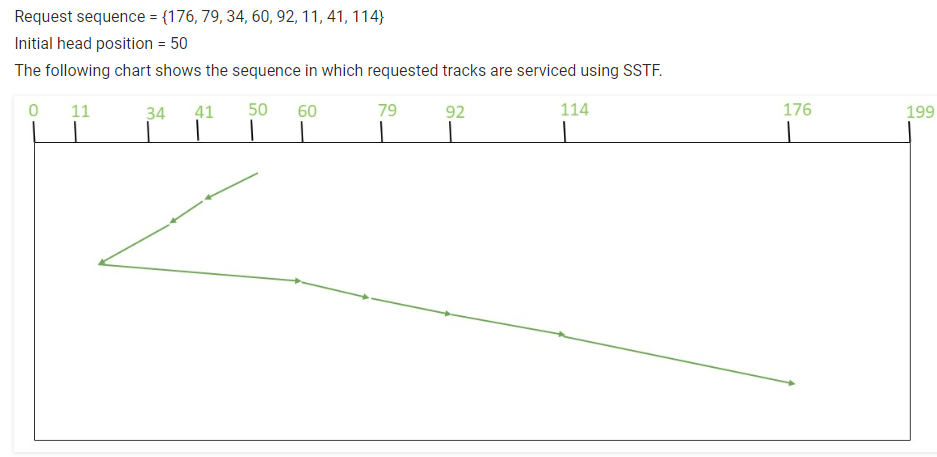
运动几次就启动几次

很多不必要的来回动

SSTF shortest seek time first



一直往绝对值最小的地方去



如果从0开始就相当于从小到大

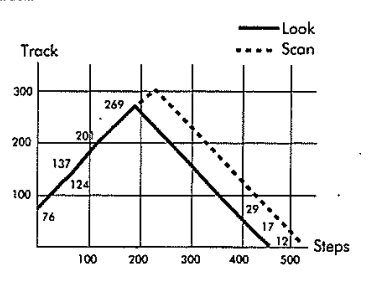
那么按这个逻辑，12 13 49 50 123 124,到50的时候插入32 45 19 6 2，那么按照逻辑，我们应该45 32 19 6 2，又回去了

3.scan and look



从start开始，一直到最大值，然后往0，然后再到新的最大值

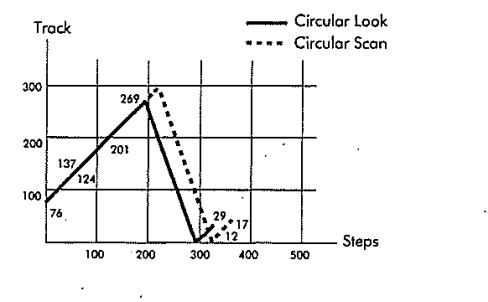




scan会到最大值，look只是到实际最大值

4.circular scan/look

增加了一个新指令，可以直接回退到0，

. 

直接最大，这样就不用花时间走回退的路

circular scan 只有在有homing的时候好于look scan

dead lock 四要素

1.要有mutual exclusion:至少要有一个resource是无法分享的状态的，同时只有一个process能进入

2.hold and wait:当其他人用这个resource的时候，这个process必须wait

3，no preemption

resource 不能被preempted强杀，得到resource的唯一方法是另外一个process完成

4. circular wait: p0在wait p1 hold的resource，p1在wait p2 hold 的resource，p2在wait p0 hold 的resource

resource allocation graph，指向resource的箭头，我需要他，request edge, resource指向process指向的箭头，这个process hold他，assignment edge

resource里的点代表一个instance、我们只需要一个点就够用了

如果没有cycle，那么就没有deadlock，如果有cycle可能有deadlock。

所以cycle是deadlock的必要非充分necessary not sufficient条件

如何避免deadlock

1.避免ME，如果文件是read only，那么多人可以进入，如果要写，那么不可能避免ME

不可能avoid or prevent deadlock

但是很多CS是互斥的  
It is not possible to dis-satisfy the mutual exclusion because some resources, such as the tap drive and printer, are inherently non-shareable.

2.hold and wait

每个进程直到做完之前，hold all process,这样就一个一个来了

缺点：low resource utilization starvation

no preemption

when process has higher priority, it can preempt resource

circular wait ,every process has special priority，

banker algorithm to avoid

algorithm detection，把resource去掉，只要有CYCLE就有dead lock，但是什么都不能做