Operating Systems exercises – Week 10 - Selected answers.

Tutorial questions

1) Suppose that a disk drive has 5000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 163, and the previous request was at cylinder 115. The queue of pending requests, in FIFO order, is 186, 2460, 513, 1764, 942, 509, 1122, 1250, 1260.

Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests, for each of the following disk scheduling algorithms?

a. FCFS
b. SSTF
c. SCAN
d. LOOK
e. C-SCAN

No answer – question and answer to be included in portfolio that is to be handed in.

2) Compare the performance of C-SCAN and SCAN scheduling, assuming a uniform distribution of requests. Consider the average response time (the time between the arrival of a request and the completion of that request's service), the variation in response time, and the effective bandwidth. How does performance depend on the relative sizes of seek time and rotational latency?

Answer: The main difference would be that C-SCAN loses time traveling from the inside edge to the outside edge of the disk. Other, quite minor differences in throughput could come from the fact that SCAN examines the edge tracks once for two examinations of the inside tracks and the time intervals between successive visits of a given track tend to have a greater variance for SCAN than for C-SCAN.

3) Explain why SSTF scheduling tends to favor middle cylinders over the innermost and outermost cylinders.

Answer: The center of the disk is the location having the smallest average distance to all other tracks. Thus after servicing the first request, the algorithm would be more likely to be closer to the center track than to any other particular track, and hence would more often go there first. Once at a particular track, SSTF tends to keep the head near this track; thus, this scheduling strategy would compound the initial tendency to go to the center.

4) Requests are not usually uniformly distributed. For example, a cylinder containing the file system FAT or inodes can be expected to be accessed more frequently than a

cylinder that only contains files. Suppose that you know that 50 percent of the requests are for a small, fixed number of cylinders.

a. Would any of the scheduling algorithms discussed in this chapter be particularly good for this case? Explain your answer.
b. Propose a disk-scheduling algorithm that gives even better performance by taking advantage of this "hot spot" on the disk.
c. File systems typically find data blocks via an indirection table, such as a FAT in DOS or inodes in UNIX. Describe one or more ways to take advantage of this indirection to improve the disk performance.

Answer:
a. SSTF would take greatest advantage of the situation. FCFS could cause unnecessary head movement if references to the "high-demand" cylinders were interspersed with references to cylinders far away.
b. One suggestion: SSTF with some modification to prevent starvation.
Second suggestion: (A variation of SCAN) The head is allowed tomake one "zigzag" over the high-volume tracks. An upper limit on the number of consecutive requests (on the same track) that can be serviced before moving on would prevent starvation.
c. No answer.

5) The fact that bad blocks are remapped by sector sparing or sector slipping could have a performance impact. Suppose that the drive in Exercise 13.17 has a total of 100 bad sectors at random locations, and that each bad sector is mapped to a spare that is located on a different track, but within the same cylinder. Estimate the I/Os per second and the effective transfer rate for a random-access workload consisting of 8-kilobyte reads, with a queue length of 1 (so the choice of scheduling algorithm is not a factor). What is the performance impact of the bad sectors?

No answer – question and answer may be used in a future portfolio.

Linux exercises.

No answers – questions and answers may be used in a future portfolio.