CS3500: OPERATING SYSTEMS -> DISK-SCHEDULING REPORT LAB7 by SAKSHAM SINGH (CS20B067)

The program used to implement the disk scheduling algorithms services a disk of 5000 cylinders. Since a 1000 random requests are being processed - which is quite high for the number of cylinders, the average head movement for most algorithms is in the range of 6 - 10. With the exception of FCFS(First Come First Serve) which gives an average head movement of about 1600-1700 to process each request.

To process the efficiency of each algorithm fairly, the initial position of the head is fixed for all - taken as the 2000th cylinder of the 5000 present (0 -4999). The abnormally high number for FCFS compared to others arises due to the fact that unlike other algorithms, it just processes the requests as they come, instead of having some knowledge of the cylinder addresses needed by each request and processing in some order which optimizes the process of fulfilling these requests.

The SSTF(Shortest Seek Time First) algorithm along with LOOK and SCAN algorithms seem to give the best performance in processing these 1000 random requests - although the SCAN algorithm seems to be slightly worse than the other two. The SSTF algorithm at each step of processing a request, chooses the one which is nearest to the head in any direction - essentially it is a greedy algorithm for processing cylinder requests which seems to work efficiently due to the randomness, and in a way uniformness of the requests. In biased or non-uniform cylinder requests, this algorithm might struggle with its performance. In the current circumstances though, it gives an average of 6.985 head movements to process each request.

The SCAN algorithm gives an average head movement of about 6.993 to process each cylinder request. From the initial head position, it goes to any one of the extremes of the disk i.e. in this case cylinder 0 or 4999, and processes any requests which come along the way. Then after reaching one extreme, it starts moving towards the other extreme, and stops after processing each request. It seems to be less efficient than the SSTF in this case, but should work better than in most non-random cases.

The LOOK algorithm is basically the same as the SCAN algorithm but it gains some efficiency in the fact that instead of going to one of the extremes, it stops when it knows that there are no more requests between the current head position and the extreme towards which it was initially moving. So, it will always be at least as efficient as, if not more than the SCAN algorithm and hence should always be preferred over it. In the given conditions, it takes about 6.98 cylinders to be traversed by the head to process each request.

The C-SCAN(Circular SCAN) and C-LOOK(Circular LOOK) algorithms works in a similar fashion to their corresponding algorithms but after reaching one extreme (or the furthest request in one direction in case of LOOK) it switches to the other extreme of the disk (or the furthest request from initial head in the other direction for C-LOOK) and then processes the requests. The complete switch from one end to other end considering the disk is helpful, but it makes the average head movement numbers to be a bit higher than their corresponding algorithms. The C-SCAN returns 9.993 and C-LOOK returns 9.994 as the average head movement to process each request.

So, the SCAN and LOOK algorithms seem to be the best and FCFS seem to be the worst.

*All the data given here is given in the folder as screenshots for each algorithm run on default head value.