22C:050 Solutions for HW#8

11.5

If one sector rotates by the head every 138 microseconds and it takes 100 microseconds for the software to start the next disk transfer after the previous one has finished, and if the head positioning latency is 0.005 seconds times the square root of the number of tracks moved, make a table of the number of sectors which must be skipped before the next transfer can begin as a function of the number of tracks the head must move, from 0 to at least 15 tracks.

$$f(tracks) = \left(\sqrt{tracks} \times 5000 + 100\right) / 138$$

Trace	Sectors skipped
0	
1	37
2	52
3	63
4	73
5	82
6	89
7	96
8	103
9	109
10	115
11	121
12	126
13	131
14	136
15	141

11.7

Figure 11.3 contains a warning that certain interface registers should not be changed while a read or write operation is in progress.

a) Describe the possible consequences of changing DISKCYL while a write operation is in progress.

DISKCYL change during a write could result in the head being moved to a different cylinder, which might result in part of the data being written on one cylinder and part of it being written on another cylinder. Thus the integrity of data can not be guaranteed.

b) Describe the possible consequences of changing DISKADDR while a read operation is in progress.

DISKADDR during a read might result in data discontinuity, overlap, or loss in memory.

11.13

Consider a disk system where the hardware allows consecutive sectors of one track to be read, but 4 sectors must be skipped to move 1 cylinder in or out, 6 must be skipped to move 2, 7 must be skipped to move 3, and 8 must be skipped to move 4. If there are 16 sectors per track (numbered 0 to 15), disk addresses are expressed as (track,sector) pairs, and requests arrive to access addresses (1,2), (2,5), (4,10), (5,14), (1,6), and (3,11), how many rotations and fractions of a rotation, at minimum, would it take to process this sequence of requests

a) if they are processed in the order they are given above?

		Read	skipped
(1,2)	(1,3)	1	
	(2,7)		4
	(2,5)		14
(2,5)	(2,6)	1	
	(4,12)		6
	(4,10)		14
(4,10)	(4,11)	1	
	(5,15)		4
	(5,14)		15
(5,14)	(5,15)	1	
	(1,6)		8
(1,6)	(1,7)	1	
	(3,13)		6
	(3,11)		14
(3,11)	(3,12)	1	

Rotations = 91/16 = 5+11/16

b) if they are processed in the best possible order?

The order is (5,14),(4,10),(2,5),(3,11),(1,2),(1,6)

		Read	skipped
(5,14)	(5,15)	1	
	(4,3)		4
	(4,10)		7
(4,10)	(4,11)	1	
	(2,1)		6
	(2,5)		4
(2,5)	(2,6)	1	
	(3,10)		4
	(3,11)		1
(3,11)	(3,12)	1	
	(1,2)		6
(1,2)	(1,3)	1	
	(1,6)		3
(1,6)	(1,7)	1	

Rotations = 41/16 = 2 + 9/16

11.14

For each of the disk scheduling policies, what should be done when a request is posted for the track the head is currently processing? What alternatives are there, if any, and what undesirable consequences would result from the use of these alternatives?

a) shortest-seek-time-first

The request should be added to the end of the FIFO queue of requests for that cylinder.

We could further order requests for the cylinder so that they could be done in the fewest rotations. But we should balance on this overhead time of processing and the gain in speed.

b) cyclic-scan

We should add the request to the queue for this cylinder after the head moves from this cylinder.

We could add it right away, which would be SSTF. However, this would cut down on fairness and increase the worse-case waiting time.

c) Elevator

The request is added to the queue for this cylinder.

An alternative is to add the request to the queue after the head move from this cylinder, which would add to fairness, but cut down on throughput.