Computer Operating systems Exam MAY 8th 1Pm-4PM

Topics

2 sections

Section A 40 questions (40mrks)

20 MCQ
20 True and false

Section B structured (practical) 60mrks

Three Questions

Q1 based on device I/O management

Which Device Scheduling Algorithm?

- FCFS works well with light loads, but as soon as load grows, service time becomes unacceptably long.
- SSTF is quite popular and intuitively appealing. It works well with moderate loads but has problem of localization under heavy loads.
- SCAN works well with light to moderate loads and eliminates problem of indefinite postponement. SCAN is similar to SSTF in throughput and mean service times.
- C-SCAN works well with moderate to heavy loads and has a very small variance in service times.

Terminology

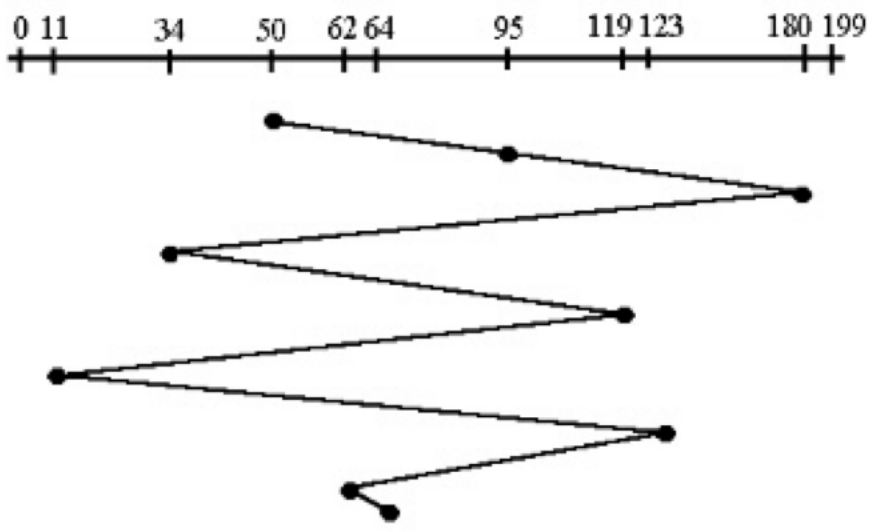
- ***access time*** – the total time required to access data in secondary storage.
- ***seek time*** – the time required to position the read/write head on the specific track from the time the I/O request is issued.
- ***rotational latency/delay*** – the time required to position a specific sector under the read/write head.
- ***transfer time*** – the time required for data to be transferred between secondary storage and main memory.

Q1 (b) disk scheduling algorithm (FCFS, SSTF, C-SCAN)

Given the following queue -- 95, 180, 34, 119, 11, 123, 62, 64 with the read-write head initially at the track 50 and the tail track being at 199 let us now discuss the different algorithms.

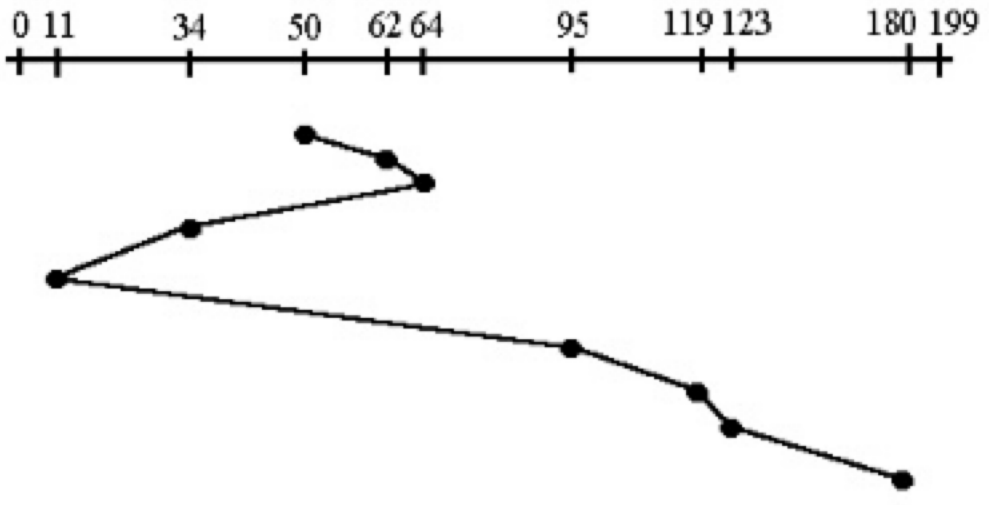
First Come -First Serve (FCFS)

All incoming requests are placed at the end of the queue. Whatever number that is next in the queue will be the next number served. Using this algorithm doesn't provide the best results. To determine the number of head movements you would simply find the number of tracks it took to move from one request to the next. For this case it went from 50 to 95 to 180 and so on. From 50 to 95 it moved 45 tracks. If you tally up the total number of tracks you will find how many tracks it had to go through before finishing the entire request. In this example, it had a total head movement of 644 tracks. The disadvantage of this algorithm is noted by the oscillation from track 50 to track 180 and then back to track 11 to 123 then to 64. As you will soon see, this is the worse algorithm that one can use.



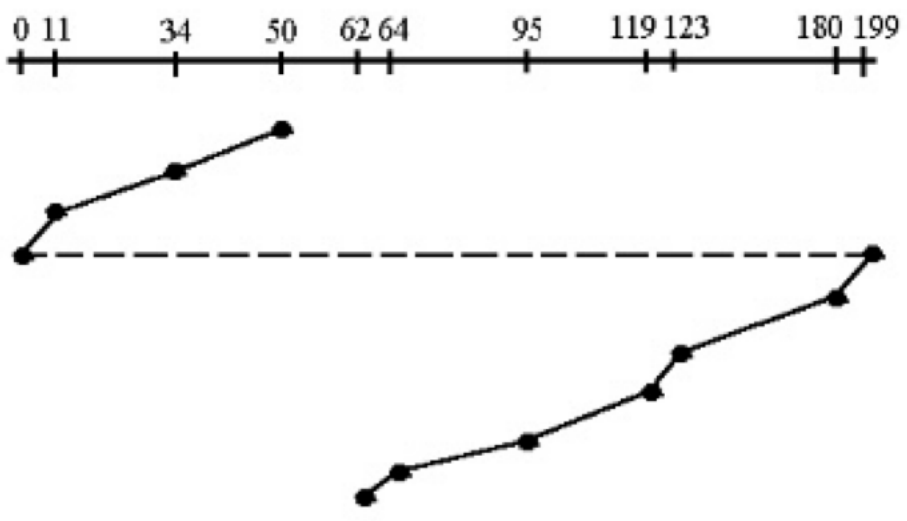
Shortest Seek Time First (SSTF)

In this case each request is serviced according to next shortest distance. Starting at 50, the next shortest distance would be 62 instead of 34 since it is only 12 tracks away from 62 and 16 tracks away from 34. The process would continue until all the process are taken care of. For example, the next case would be to move from 62 to 64 instead of 34 since there are only 2 tracks between them and not 18 if it were to go the other way. Although this seems to be a better service being that it moved a total of 236 tracks, this is not an optimal one. There is a great chance that starvation would take place. The reason for this is if there were a lot of requests close to each other the other requests will never be handled since the distance will always be greater.



Circular Scan (C-SCAN)

Circular scanning works just like the SCAN to some extent. It begins its scan toward the nearest end and works its way all the way to the end of the system. Once it hits the bottom or top it jumps to the other end and moves in the same direction. Keep in mind that the huge jump doesn't count as a head movement. The total head movement for this algorithm is only 187 tracks, but still this isn't the most efficient.



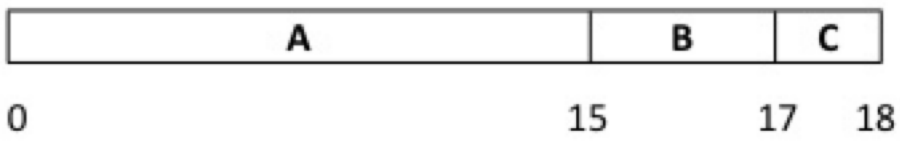
Q2 Process Scheduling

(a) Practical Choose one pre-emptive and one non pre-emptive

(B) calculate turnaround time and average

Non pre-emptive FCFS

Timeline/Gantt chart - FCFS



Turnaround time = completion time - arrival time

A: 15 – 0 = 15

B: 17 – 0 = 17

C: 18 – 0 = 18

Average turnaround time = (15 + 17 + 18)/3 = 16.67 ms

Preemptive SRT

Timeline/Gantt chart – SRT (pre-emptive)

Arrival time:	0	1	2	3
Job:	A	B	C	D
CPU Cycle:	6	3	1	4
	5	2	F	F
		F		

Turnaround time:

A: 14 – 0 = 14

B: 5 – 1 = 4

C: 3 – 2 = 1

D: 9 – 3 = 6

Avg turnaround time: (14 + 4 + 1 + 6)/4 = 6.25ms

(C) Theory

Scheduling criteria

A Good Scheduling Policy

- *Maximize throughput* by running as many jobs as possible in a given amount of time.
- *Maximize CPU efficiency* by keeping CPU busy 100 % of time.
- *Ensure fairness for all jobs* by giving everyone an equal amount of CPU and I/O time.
- *Minimize response time* by quickly turning around interactive requests.
- *Minimize turnaround time* by moving entire jobs in/out of system quickly.
- *Minimize waiting time* by moving jobs out of READY queue as quickly as possible.

Q3 Page replacement policies

(a) Trace table FIFO and LRU

EG

5. Trace Table Using FIFO

Request	A	B	A	C	A	B	D	B	A	C	D
Frame 1	A	A	A	A	A	A	D	D	D	D	D
Frame 2	empty	B	B	B	B	B	B	B	A	A	A
Frame 3	empty	empty	empty	C	C	C	C	C	C	C	C
interrupt	*	*	*	*			*		*		

Failure Rate (no. of interrupts/pages) = 6/11 * 100 = 54.54ʳ %

Success rate (no. of uninterrupted frames/pages) = 5/11 * 100 = 45.45ʳ %

b. Trace Table Using LRU (Least Recently Used)

Request	A	B	A	C	A	B	D	B	A	C	D
Frame 1	A	A	A	A	A	A	A	A	A	A	A
Frame 2	empty	B	B	B	B	B	D	D	D	C	C
Frame 3	empty	empty	empty	C	C	C	C	B	B	B	D
interrupt	*	*	*	*		*	*	*		*	*

Failure Rate (no. of interrupts/pages) = 9/11 * 100 = 81.8ʳ %

Success rate (no. of uninterrupted frames/pages) = 2/11 * 100 = 18.18ʳ %