CS303 | Assignment-4

Design Decisions:

- For storing requests data, struct is created which contains platter number, cylinder number, sector number and number of sectors to be accessed.
  ```
  struct request_data{    //structure of requests
      int platter;
      int cylinder;
      int sector;
      int number;
  };
  ```
- For storing position of the read write head, a struct is created which stores which track the read write head is at and what sector it is currently at.
  ```
  struct rwarm{ //structure of rwhead
    int cylinder;
    int sector;
   };
  ```
- Random scheduling is implemented by using qsort function in C and using a random comparator function as one of its arguments.
  ```
  qsort((void*)res, rs, sizeof(res[0]), rand_comparator);
  ```
- For FIFO, the order of created requests is not changed.
- For SCAN, the request is ordered in such a way that all requests greater than or equal to track number 12 appear first in increasing order and then decrease in order to the lowest track number.
- For C-SCAN, requests are arranged in increasing order on the basis of their track number starting from 12 and then those requests with track number < 12 are inserted in increasing order.
- In SSTF, initially rwhead is at 12 and the nearest track to it is selected and added to request array and then the rwhead is updated to point to the selected value and the nearest to it is selected and added to the request array and this process continues till all the requests are ordered.
- The program takes in paramters like rotations per minute, average access time, sector size, number of requests and which algorithm to use.
- Requests are generated in random manner and stored in an array of request struct.
- After the requests are created, '*disk*' function is called which then selects a suitable ordering algorithm to order the requests according to user input and then calculated response time for the requests.
- Response time is calculated as:

response time[i] = response time[i-1]
+ abs{current head - requested head} * time cross 1 track
+ min(abs{current sector - requested sector}, 20 - abs{current sector - requested sector}) * time to cross 1 sector

+ (number of sectors -1)* time cross 1 sector

with base case of response[i] = 0, i<=0.

Parameters : r, N , T_s	Scheduling policy	Requests	Throughput (requests/s)	<Avg., Min., Max., StdDev.> of response time (ms)
7500, 512, 4ms	Random	Sequence of randomly generated addresses: <platter, cylinder, sector>	102.2331	Avg response: 49001.398438 ms min response: 7.600000 ms max response: 97815.648438 ms StdDev time: 28164.898438 ms
15000, 512, 4ms	Random	Sequence of randomly generated addresses: <platter, cylinder, sector>	145.150848	Avg response: 34517.167969 ms min response: 3.800000 ms max response: 68893.843750 ms StdDev time: 19818.837891 ms
7500, 512, 4ms	FIFO	Sequence of randomly generated addresses: <platter, cylinder, sector>	102.574928	Avg response: 48762.644531 ms min response: 7.200000 ms max response: 97489.710938 ms StdDev time: 28192.095703 ms
15000, 512, 4ms	FIFO	Sequence of randomly generated addresses: <platter, cylinder, sector>	145.678925	Avg response: 34294.042969 ms min response: 3.600000 ms max response: 68644.109375 ms StdDev time: 19846.332031 ms
7500, 512, 4ms	SSTF	Sequence of randomly generated addresses:	173.414597	Avg response: 28784.410156 ms min response: 7.200000 ms

		<platter, cylinder, sector>		max response: 57665.269531 ms StdDev time: 16620.691406 ms
15000, 512, 4ms	SSTF	Sequence of randomly generated addresses: <platter, cylinder, sector>	346.728149	Avg response: 14396.431641 ms min response: 3.600000 ms max response: 28841.039062 ms StdDev time: 8313.219727 ms
7500, 512, 4ms	SCAN	Sequence of randomly generated addresses: <platter, cylinder, sector>	172.908340	Avg response: 28931.328125 ms min response: 7.200000 ms max response: 57834.113281 ms StdDev time: 16680.289062 ms
15000, 512, 4ms	SCAN	Sequence of randomly generated addresses: <platter, cylinder, sector>	345.710449	Avg response: 14469.939453 ms min response: 3.600000 ms max response: 28925.939453 ms StdDev time: 8343.208008 ms
7500, 512, 4ms	C-SCAN	Sequence of randomly generated addresses: <platter, cylinder, sector>	173.391785	Avg response: 28903.718750 ms min response: 7.200000 ms max response: 57672.859375 ms StdDev time: 16643.277344 ms
15000, 512, 4ms	C-SCAN	Sequence of randomly generated addresses: <platter, cylinder, sector>	346.647949	Avg response: 14457.218750 ms min response: 3.600000 ms max response: 28847.710938 ms StdDev time: 8325.737305 ms

OBSERVATIONS

Based on the response time statistics, it can be concluded that random and fifo have almost equal average response time, with max, min and standard deviation also approximately same. They have the highest average response time when comparred against other disk scheduling algorithms.

SSTF, SCAN and C-SCAN have approximately same statistics, with the order amongst them as lowest is sstf, followed by c-scan and then scan.

It can be explained that fifo has greater response time than sstf, scan and c-scan as in fifo, the read write head needs to move more due to increased number of forward and backward movement, whereas in sstf, scan and c-scan, movement of read write head among the tracks is reduced.

Also, it is observed that as rpm increases throughput increases and response time decreases overall. This trend is observed for all the algorithms. This can be explained as if rate of rotation is increased, then more requests can be fulfilled per unit time, increasing throughput and decreasing response time.