**Viktoryia Strylets 111748510**

**Device Management statics Analysis**

The statistics show the difference in the performance of the two scheduling algorithms for sequence of I/O requests - FIFO which is used in the Demo.jar and (modified) MSSTF (which uses multiple queues, this approach reduces the possibility of starvation of I/O requests and improves overall performance of the algorithm) in the OSP.jar

Note! The numbers stated in the comparison are the averaged statistics of the all snapshots produced in the log files of Osp .jar and Demo.jar.

The statistics of the interests of

Demo.jar using **FCFS:** FCFS is the simplest of all the Disk Scheduling Algorithms. In FCFS, the requests are addressed in the order they arrive in the disk queue.

|  |  |  |  |
| --- | --- | --- | --- |
| DEVICES Demo.jar : | Total number of tracks swept on each device | Average number of tracks swept per I/O request. | Average turnaround time for I/O requests. |
| Device 0 | 5215 | 6 | 645 |
| Device 1 | 71.4 | 2 | 252 |
| Device 2 | 20 | 1 | 770 |

The statistics of Osp.jar using (**MSSTF**):

The Modified shortest-service-time-first policy, it uses SSFT algorithm in the each queue to select the disk I/O request that requires the least movement of the disk arm from its current position in the specific queue. Thus, we always choose to incur the minimum seek time. Moreover, its use multiple queues to store requests and regularly switch between the queues to process requests, this strategy reduces the possibility of starvation and increases the average turnaround time compare to the SSTF.

|  |  |  |  |
| --- | --- | --- | --- |
| DEVICES: OSP.jar | Total number of tracks swept on each device | Average number of tracks swept per I/O request. | Average turnaround time for I/O requests. |
| Device 0 | 1743 | 2 | 637 |
| Device 1 | 57.5 | 1 | 432 |
| Device 2 | 16.6 | 1 | 802 |

**FCFS:**

Advantages over MSSTF:

* Every request gets a fair chance
* Better Performance **in the Average turnaround time for I/O requests**, because the I/O requests are scheduled in the order they arrive and the time between its arriving and completion are minimized.
* No indefinite postponement

Disadvantages over MMSTF:

* Does not try to optimize seek time
* **The Average number of the swept tracks** are larger than in the MSSFT due to not optimized seek time.
* **The System Throughput** is much less compared to the MSSTF.May not provide the best possible service

**MSSTF:**

MSSTF is certainly an improvement over FCFS as it decreases the average response time and increases the throughput of system.

Advantages:

* **Average Response Time decreases**
* **Throughput increases**
* **The Average number of the swept track** are less because from each queue it selects the disk I/O request that requires the least movement of the disk access arm from its current position regardless of direction

Disadvantages:

* Can cause Starvation for a request if it has higher seek time as compared to incoming requests, but compare to the regular SSTF algorithm, the possibility of starvation is reduced, due to the multiple queues SSTF scheduling.
* High variance of response time as SSTF favors only some requests