SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES, CHENNAI – 602 105

PROJECT REPORT

TITLE

DISK I/O SCHEDULER SIMULATION EVALUATING PERFORMANCE

Submitted to

SAVEETHA SCHOOL OF ENGINEERING

By

Name1: HARSHA VARDHAN(192124074)

Name2: KALYAN. K (192110258)

Name3: SWATHI(192110634)

Guided by

Dr.G.Mary Valentina

Abstract:

Disk scheduling plays a critical role in optimizing data access on modern storage devices. This project aims to implement and evaluate various disk scheduling algorithms to understand their performance characteristics under different workloads. The study encompasses widely used algorithms such as First Come First Serve (FCFS), Shortest Seek Time First (SSTF), SCAN, C-SCAN, LOOK, and C-LOOK.The project begins with the Implementation of these algorithms in a simulated environment, considering factors such as head movement, seek time, and request queues. Each algorithm is meticulously coded and tested to ensure accuracy and efficiency in disk access.Furthermore, a comparative analysis is conducted to evaluate the performance of these algorithms under diverse scenarios, including varying request patterns and workload intensities. Metrics such as average seek time, throughput, and fairness are utilized to assess the effectiveness of each scheduling algorithm.Additionally, the project explores the practical implications of these algorithms in real-world scenarios, considering factors like disk fragmentation, system load, and concurrency. Insights gained from this study can aid system designers and administrators in selecting the most appropriate disk scheduling algorithm based on specific application requirements and resource constraints.In summary, this project contributes to a deeper understanding of disk scheduling algorithms and provides valuable insights into their practical implications for system performance and efficiency**.**

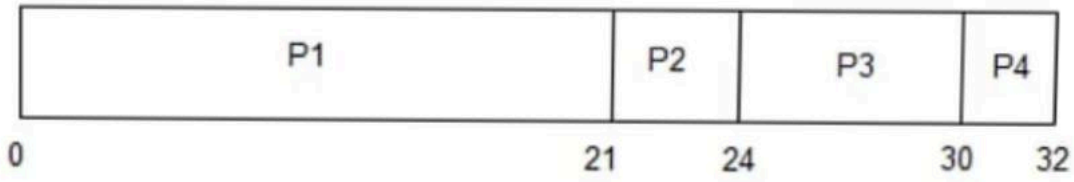
Introduction:

In the realm of computer systems, efficient data access and management are paramount for optimal performance. One critical aspect of this is disk scheduling, which determines the order in which read and write requests are serviced on storage devices such as hard disk drives (HDDs) and solid-state drives (SSDs). Disk scheduling algorithms play a vital role in minimizing

seek time, reducing latency, and improving overall system throughput.The object’ve of this project is to delve into the realm of disk scheduling algorithms, exploring their mechanisms, implementations, and performance characteristics. By comprehensively studying these algorithms, we aim to gain insights into their strengths, weaknesses, and suitability for different system environments and workloads.The project begins with an overview of the importance of disk scheduling in modern computing systems. As data storage demands continue to grow exponentially, efficient disk access becomes increasingly critical for ensuring smooth operation and optimal resource utilization. Disk scheduling algorithms serve as the bridge between high-level application requests and low-level disk operations, orchestrating data movement to minimize access times and maximize throughput.Following the introduction, we delve into the core concepts and principles underlying disk scheduling algorithms. We explore fundamental metrics such as seek time, rotational latency, and transfer time, which influence the performance of disk scheduling decisions. Additionally, we examine various scheduling policies and strategies employed by different algorithms to organize and service disk requests effectively.The project then proceeds to implement several prominent disk scheduling algorithms in a simulated environment. These include classic algorithms such as First Come First Serve (FCFS), Shortest Seek Time First (SSTF), SCAN, C-SCAN, LOOK, and C-LOOK. Each algorithm is meticulously coded and tested to evaluate its behavior under different scenarios and workloads.Furthermore, we conduct a comparative analysis of the implemented algorithms, examining their performance across a range of benchmarks and synthetic workloads. Metrics such as average seek time, throughput, and fairness are utilized to assess the efficacy of each scheduling algorithm under varying conditions.In addition to performance evaluation, the project also considers practical considerations and trade-offs associated with disk scheduling.

PROCESS	BURST TIME
P1	21
P2	3
P3	6
P4	2

The average waiting time will be = (0 + 21 + 24 + 30)/4 = 18.75 ms



This is the GANTT chart for the above processes

Process:

The process for a disk scheduling project typically involves several key steps, including research, implementation, testing, evaluation, and documentation. Here's a detailed breakdown of each step in the process

Research and Understanding: Begin by researching the fundamentals of disk scheduling algorithms. Understand the key concepts such as seek time, rotational latency, disk arm movement, and scheduling policies.Study various disk scheduling algorithms, including FCFS, SSTF, SCAN, C-SCAN, LOOK, and C-LOOK. Gain insights into their advantages, disadvantages, and suitability for different scenarios.Review existing literature, academic papers, and textbooks on

operating systems and storage systems to deepen your understanding of disk scheduling principles.

Algorithm Selection: Based on your research, select the disk scheduling algorithms you plan to implement and evaluate in your project.Consider factors such as algorithm complexity, performance characteristics, and relevance to your project goals.

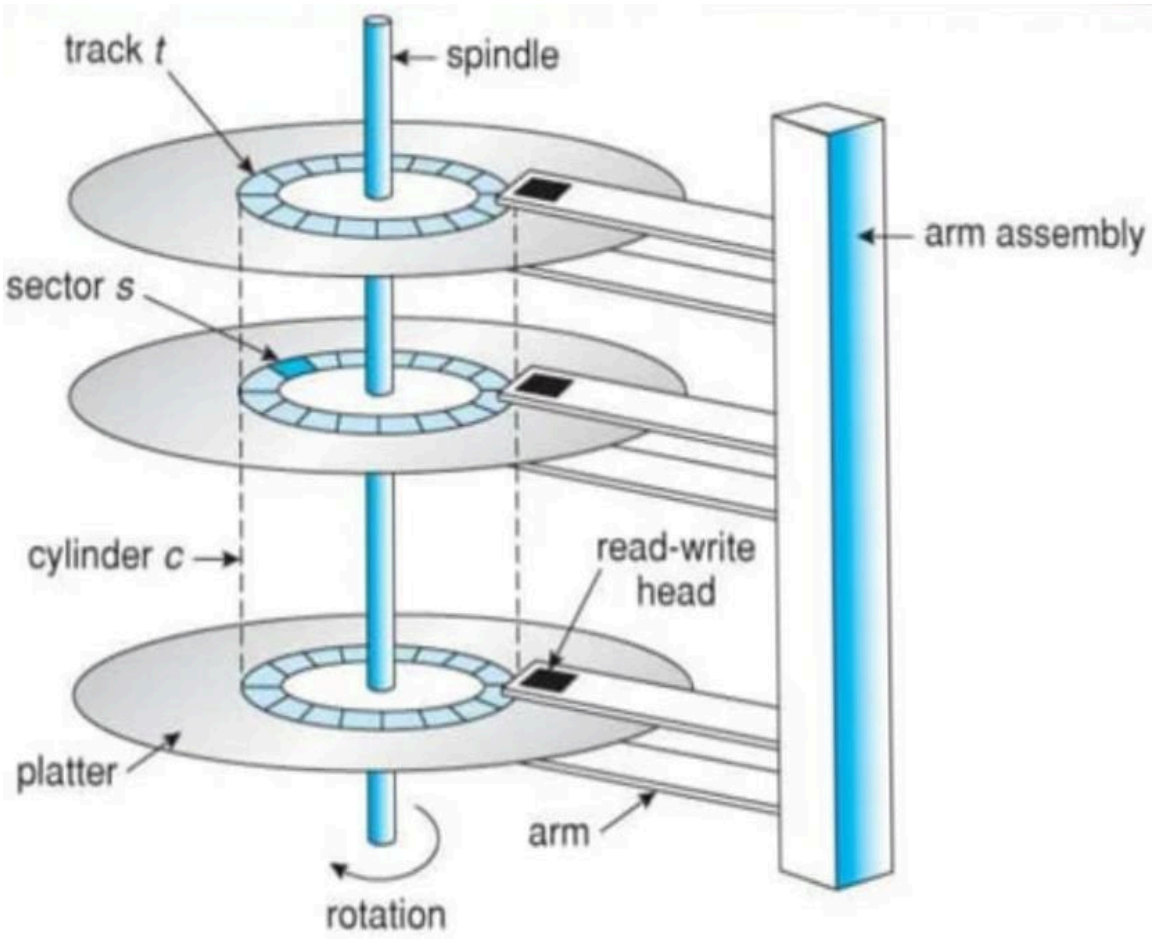
Implementation: Implement the selected disk scheduling algorithms in your chosen programming language (e.g., C, C++, Python).Ensure that your implementations accurately reflect the behavior and logic of each algorithm.Develop data structures and algorithms to simulate disk requests, track disk arm movement, and measure performance metrics.

Testing and Validation: Test your implementations rigorously using various test cases and workload scenarios.Verify that each algorithm behaves as expected and produces correct results.Conduct unit tests, integration tests, and system tests to identify and rectify any bugs or inconsistencies.

Performance Evaluation: Design experiments to evaluate the performance of each disk scheduling algorithm.Define performance metrics such as average seek time, throughput, response time, and fairness.Run experiments using synthetic workloads and real-world traces to gather performance data for analysis.

Comparative Analysis: Compare the performance of different disk scheduling algorithms using the collected data.Identify strengths and weaknesses of each algorithm in terms of performance metrics.Analyze trade-offs between algorithms in terms of complexity, fairness, and adaptability to workload variations.

Documentation and Reporting: Document your implementation details, including data structures, algorithms, and design decisions.Summarize your experimental setup, including test scenarios, workloads, and performance metrics.Present your findings in a clear and organized manner, including tables, charts, and graphs to illustrate results.



Objective:

The objectives for a disk scheduling project can vary based on the scope and goals of the study. However, here are some common objectives that you might consider for your disk scheduling project:

Implement Disk Scheduling Algorithms: Develop implementations of various disk scheduling algorithms, including classic algorithms like FCFS, SSTF, SCAN, C-SCAN, LOOK, and C-LOOK.

Performance Evaluation: Evaluate the performance of implemented disk scheduling algorithms under different workloads, considering metrics such as average seek time, throughput, response time, and fairness.

Comparative Analysis: Conduct a comparative analysis of the implemented disk scheduling algorithms to identify their strengths, weaknesses, and suitability for different system environments and workload patterns.

Optimization Strategies: Explore optimization strategies for disk scheduling algorithms to improve performance, reduce latency, and enhance system throughput.

Real-world Application: Investigate the practical implications of disk scheduling algorithms in real-world scenarios, considering factors such as disk fragmentation, system load, and concurrency.

Algorithmic Trade-offs: Analyze the trade-offs between different disk scheduling algorithms in terms of complexity, adaptability, and efficiency, providing insights into the advantages and limitations of each approach.

Resource Utilization: Evaluate the impact of disk scheduling algorithms on resource utilization, including CPU usage, disk utilization, and system responsiveness.

Scalability and Robustness: Assess the scalability and robustness of disk scheduling algorithms under varying system loads, disk capacities, and concurrency levels.

Future Directions: Identify potential areas for further research and development in disk scheduling algorithms, suggesting avenues for innovation and improvement in storage system optimization techniques.

Documentation and Dissemination: Document the project methodology, experimental setup, findings, and conclusions in a comprehensive report or research paper, contributing to the body of knowledge in the field of disk scheduling and operating systems.

Literature review :

A literature review on disk scheduling would typically cover various theoretical aspects, including the principles of disk scheduling algorithms, their

classifications, historical development, and comparative analyses. Here's a structured outline for a literature review on disk scheduling theory:

Introduction to Disk Scheduling: Define disk scheduling and its importance in computer systems.Briefly discuss the role of disk scheduling algorithms in optimizing data access on storage devices.

Fundamental Concepts: Explain the fundamental components of disk access, including seek time, rotational latency, and transfer time.Discuss how these components contribute to the overall disk access time and influence the design of disk scheduling algorithms.

Disk Scheduling Algorithms: Provide an overview of common disk scheduling algorithms, such as:

First Come First Serve (FCFS)

Shortest Seek Time First (SSTF)

SCAN

C-SCAN

LOOK

C-LOOK

Classifications of Disk Scheduling Algorithms: Classify disk scheduling algorithms based on their scheduling policies, such as:

Non-anticipatory vs. anticipatory algorithms

Deterministic vs. probabilistic algorithms

Static vs. dynamic algorithms

Discuss how these classifications influence the behavior and performance of disk scheduling algorithms.

Historical Development:

Trace the historical development of disk scheduling algorithms, starting from early disk scheduling policies to modern approaches.

Highlight key milestones, influential research papers, and significant advancements in disk scheduling theory and practice.

Comparative Analysis:

Review existing comparative studies and evaluations of disk scheduling algorithms.

Summarize findings regarding the performance, efficiency, and suitability of different algorithms under various workloads and system conditions.

Identify trends, patterns, and areas of consensus or divergence among researchers.

Challenges and Open Issues:

Discuss challenges and open issues in disk scheduling research, such as:

Addressing performance bottlenecks in high-concurrency and high-throughput environments

Dealing with dynamic workload patterns and unpredictable access patterns

Integrating disk scheduling algorithms with other system components, such as file systems and storage management layers

Future Directions and Research Opportunities:

Explore potential avenues for future research and innovation in disk scheduling theory and practice.

Suggest areas for improvement, novel algorithmic approaches, and interdisciplinary collaborations to advance the state-of-the-art in disk scheduling.

Conclusion:

Summarize key findings from the literature review.

Highlight the importance of disk scheduling in modern computing systems and the ongoing relevance of research in this field.

References:

Provide a comprehensive list of references cited throughout the literature review, including academic papers, books, and other scholarly sources.

By structuring your literature review according to these categories, you can provide a thorough overview of disk scheduling theory, research trends, and future directions in the field.

Conclusion:

In conclusion, disk scheduling plays a crucial role in optimizing data access on storage devices within computer systems. Through this literature review, we have explored the theoretical underpinnings, historical development, and comparative analysis of disk scheduling algorithms. Here are the key takeaways:

Importance of Disk Scheduling: Disk scheduling algorithms are essential for minimizing seek time, reducing latency, and maximizing throughput when accessing data stored on disk drives. By orchestrating the order in which disk requests are serviced, these algorithms significantly impact system performance and efficiency.

Variety of Algorithms: We have examined a variety of disk scheduling algorithms, including classic approaches such as FCFS, SSTF, SCAN, C-SCAN, LOOK, and C-LOOK. Each algorithm employs unique scheduling policies and strategies to optimize disk access under different scenarios and workload patterns.

Classification and Comparison: Disk scheduling algorithms can be classified based on their scheduling policies, determinism, and adaptability. Comparative studies have evaluated the performance of these algorithms across various metrics, shedding light on their relative strengths, weaknesses, and suitability for different system environments.

Historical Development: The historical development of disk scheduling algorithms reflects a progression from simplistic policies to more sophisticated approaches capable of handling diverse workload characteristics. Milestones in research and significant advancements have contributed to the evolution of disk scheduling theory and practice.

Challenges and Future Directions: Despite significant progress, challenges remain in optimizing disk scheduling for modern computing systems. Addressing issues such as high-concurrency workloads, dynamic access patterns, and integration with other system components poses ongoing research opportunities. Future directions in disk scheduling research may involve exploring adaptive algorithms, machine learning techniques, and hybrid approaches to address emerging challenges.

In summary, disk scheduling continues to be a vibrant area of research with practical implications for system performance, efficiency, and reliability. By advancing our understanding of disk scheduling theory and exploring innovative algorithmic approaches, researchers can contribute to the ongoing evolution of storage system optimization techniques, ultimately enhancing the capabilities of modern computing environments.

Reference:

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts (10th ed.). Wiley.
2. Tanenbaum, A. S., & Bos, H. (2014). Modern Operating Systems (4th ed.). Pearson.
3. Stallings, W. (2014). Operating Systems: Internals and Design Principles (8th ed.). Pearson.
4. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne. (2005). Operating System Concepts, 7th Edition. John Wiley & Sons.
5. Arpaci-Dusseau, R. H., & Arpaci-Dusseau, A. C. (2014). Operating Systems: Three Easy Pieces. Arpaci-Dusseau Books.
6. Kang, J., & Molina, C. (2005). Disk Scheduling in Multimedia Operating Systems: A Survey. ACM Computing Surveys (CSUR), 37(4), 316-344.

7. Gibson, G. A., Nagle, D. F., Amiri, K., Chang, F., Feinberg, E., Gobioff, H., … & Zeldovich, N. (1998). File system workload on a scientific multiprocessor. ACM SIGMETRICS Performance Evaluation Review, 26(1), 29-40.
8. Anderson, E., Fedorova, A., Weldon, C., Ganger, G. R., & McKusick, M. K. (2001). Scheduler activations: effective kernel support for the user-level management of parallelism. ACM Transactions on Computer Systems (TOCS), 19(2), 117-152.
9. Solomon, M. G., Russinovich, M. E., & Ionescu, A. (2016). Windows Internals, Part 1: System architecture, processes, threads, memory management, and more (7th ed.). Microsoft Press.
10. Smith, J. E., & Nair, R. (2005). Virtual Machines: Versatile Platforms for Systems and Processes. Morgan Kaufmann.
11. These references cover various aspects of disk scheduling, including theoretical foundations, practical implementations, research surveys, and real-world case studies, providing a comprehensive understanding of the topic.