Assignment Web Similarity Analysis

Generated on 2025-03-26 05:16:25

Executive Summary

Overall Web Similarity Score: 25%

Assessment: The assignment demonstrates some similarity to online resources, primarily in the descriptions of Round Robin scheduling. However, the majority of the assignment focuses on the student's specific implementation and analysis, which appears original. The similarities are mostly conceptual explanations and common phrases used when discussing scheduling algorithms.

Conclusion: While the assignment uses common terminology and explains standard scheduling algorithms, the specific implementation details, output figures, and analysis of average waiting times suggest original work. The similarities found relate to general descriptions of Round Robin scheduling, which are widely available online and considered common knowledge in operating systems courses. Therefore, the assignment does not exhibit strong evidence of plagiarism. However, attributing the sources of the common knowledge descriptions of the algorithms would be good academic practice.

Web Sources Analyzed

Source URL	Similarity Score
https://www.geeksforgeeks.org/round-robin-scheduling-with-different-arrival-times/	53.55%
https://www.geeksforgeeks.org/program-for-round-robin-scheduling-for-the-same-arrival-time/	54.81%
https://en.wikipedia.org/wiki/Round-robin_scheduling	65.09%
https://www.gatevidyalay.com/round-robin-round-robin-scheduling-examples/	57.21%

Detailed Content Matches

Match 1 - Similar Content (70%)

Assignment: Looking at the round robin algorithm which has fixed quantum of time to every process.

Source: https://www.gatevidyalay.com/round-robin-round-robin-scheduling-examples/

Source Text: In Round Robin Scheduling,CPU is assigned to the process on the basis of FCFS for a fixed amount of time. This fixed amount of time is called as time quantum or time slice.

Match 2 - Similar Content (75%)

Assignment: That means if one of the process unable to finish the process within the time slot it will be removed from the queue and put it to the last in order of the queue.

Source: https://www.gatevidyalay.com/round-robin-round-robin-scheduling-examples/

Source Text: After the time quantum expires, the running process is preempted and sent to the ready queue. Then, the processor is assigned to the next arrived process.

Match 3 - Common Knowledge (90%)

Assignment: Round Robin Scheduling

Source: https://www.geeksforgeeks.org/round-robin-scheduling-with-different-arrival-times/

Source Text: Round Robin Scheduling Algorithm

Match 4 - Common Knowledge (100%)

Assignment: Round Robin Scheduling
Source: https://www.geeksforgeeks.org/program-for-round-robin-scheduling-for-the-same-arrival-time/
Source Text: Round Robin Scheduling

Match 5 - Common Knowledge (95%)

Assignment: Round robin scheduling
Source: https://en.wikipedia.org/wiki/Round-robin_scheduling
Source Text: Round-robin scheduling

Match 6 - Common Knowledge (100%)

Assignment: Round Robin Scheduling
Source: https://www.gatevidyalay.com/round-robin-round-robin-scheduling-examples/
Source Text: Round Robin Scheduling

Full Assignment with Highlighted Plagiarism

Sections highlighted in yellow with red text indicate potential plagiarism.

EE6253 – Operating Systems and Network Programming

TAKE HOME ASSIGNMENT

EG/2021/4432: BANDARA KMTON

Implement the FCFS scheduling algorithm

Figure 1: Function for Implementation FCFS scheduling algorithm

Figure 1: Function for Implementation FCFS scheduling algorithm

Implement the SJF scheduling algorithm

Figure 2: Function for implementing SJS Scheduling algorithm

Figure 2: Function for implementing SJS Scheduling algorithm

Implement the Round Robin (RR) scheduling algorithm with a time quantum of 4 units.

Figure 3: Implement of Round Robin (RR) scheduling algorithm with a time quantum of 4 units

Figure 3: Implement of Round Robin (RR) scheduling algorithm with a time quantum of 4 units

Implement the Priority Scheduling algorithm (lower number indicates higher priority).

Figure 4: Implement the Priority Scheduling algorithm (lower number indicates higher priority).

Figure 4: Implement the Priority Scheduling algorithm (lower number indicates higher priority).

Execution of outputs

g

Figure 8: Priority Scheduling

Figure 8: Priority Scheduling

Figure 7: Round Robin Scheduling

Source: https://www.gatevidyalay.com/round-robin-round-robin-scheduling-examples/

Figure 7: Round Robin Scheduling

Source: https://www.gatevidyalay.com/round-robin-round-robin-scheduling-examples/

Figure 6 : SJF Scheduling

Figure 6 : SJF Scheduling

Figure 5: FCFS Scheduling

Figure 5: FCFS Scheduling

Comparison of average waiting time for each algorithm.

Figure 9: Comparison of average waiting time for each algorithm

Figure 9: Comparison of average waiting time for each algorithm

Write a brief analysis comparing the performance of the four scheduling algorithms based on the average waiting time.

Considering one by one of these algorithms as figure 5 shown as the output of the FCFS scheduling here the average waiting time is 7.20. So this is executing as these processes in order when they arrived. Considering that when P1 starts and after finishing of the P1 then P2 starting so this is called as non-pre-emptive. This has a higher average waiting value because of the convey effect.

Then considering about the SJS scheduling algorithm it takes the average waiting time as 4.60. comparing with FCFS algorithm that time is a lower value. Considering the process of here it will be choosing the burst value which has a lower value from the queue which will be ready to proceed. Considering the reason for having shortest value for this one is it will be going to prioritize start with the shortest processes other than the processes has a longtime slot.

Looking at the round robin algorithm which has fixed quantum of time to every process. That means if one of the process unable to finish the process within the time slot it will be removed from the queue and put it to the last in order of the queue. But considering the above algorithms as FCFS and SJS average times round robin algorithm has a highest average time because the reason for this is time quantum. But this process which ensure that there is no process wait.

Source: https://www.gatevidyalay.com/round-robin-round-robin-scheduling-examples/

By considering at the behaviour of Priority scheduling which will be going to prioritise the each process which mean if the process has highest priority it will be going to execute first. So consider about the average time of the Priority algorithm has given as 8.20 which is larger than FCFS and SJS .if the prioritizing process have the short time period then the waiting time is become low. In here can be happened the starvation because of having the delay of the low priority processes. So considering this processes mainly depend on the prioritizing of the processes and can have the starvation for the low processes.

So considering those four algorithms SJS is the most efficient because of having minimum waiting time. In FCFS can have the longest time period because of the long processes arriving first.and also for the round robin process is only depend on the time quantizing with ensuring the processes not wait too long. For the general usage the SJF is the better one having minimum time and also Round robin is ideal for ensuring the processes procedures.

Analysis Methodology

Web Similarity Analysis Method: This report analyzes the similarity between a student assignment and web content using multiple approaches:

1. **Basic similarity analysis** using TF-IDF vectorization and cosine similarity metrics to calculate statistical similarity between texts.
2. **Advanced semantic analysis** using Google's Gemini AI to identify conceptual similarities, common phrases, and potential plagiarism patterns.
3. **Source verification** by analyzing multiple sources to distinguish between common knowledge and unique content.

Interpretation Guide:

- 0-15%: Very low similarity - Likely original content
- 16-30%: Low similarity - Contains common phrases but largely original
- 31-50%: Moderate similarity - May contain some paraphrased content
- 51-70%: High similarity - Contains substantial similar content
- 71-100%: Very high similarity - Significant portions may be unoriginal

---

Disclaimer: This automated similarity analysis provides an approximation of content similarity against web sources. Results should be interpreted by a human reviewer for context-appropriate assessment. Common knowledge, standard phrases, and coincidental matches may be flagged and require human judgment.