22CSCI07I Assignment 1: Centralized Mutual Exclusion Algorithm

AHMED RAMIE

*217380*

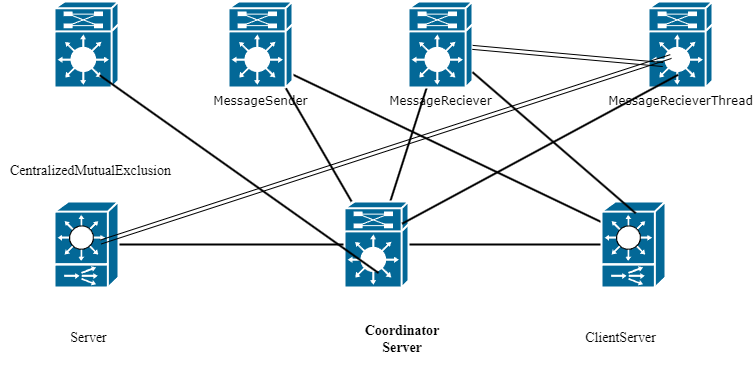
|  |  |  |
| --- | --- | --- |
| Name | ID | EMAIL |
| **Ahmed** **Ramie** | **217380** | [Ahmed217380@bue.edu.eg](mailto:Ahmed217380@bue.edu.eg) |

Author Information:

Introduction

* Centralized Mutual Exclusion Algorithm (CMEA) is a token-based approach used to achieve mutual exclusion in distributed systems. In this algorithm, there is a central server that manages access to the shared resource. The server issues a token to a process that requests access to the resource. A process can only access the resource if it holds the token. If a process wants to access the resource and the token is currently held by another process, it must wait until the token is released.
* The CMEA algorithm ensures that only one process can access the shared resource at a time. The server is responsible for maintaining the token and keeping track of which process is currently holding the token. When a process releases the token, the server checks if there are any other processes waiting for the token and passes it on to the next process in the queue.
* The advantage of the CMEA algorithm is that it is simple and easy to implement. However, it also has some drawbacks, such as a single point of failure - if the central server fails, the entire system will be affected. Moreover, the system may experience high latency due to the need to wait for the token to be released.
* Overall, the CMEA algorithm is a basic approach to achieving mutual exclusion in distributed systems. Its simplicity makes it a popular choice for many applications, but its drawbacks must be considered when designing systems for high availability and low latency.

Diagram



Advantages and disadvantages

Advantages:

1. Simplicity: This algorithm is relatively simple to implement and understand as compared to other algorithms for mutual exclusion.
2. Efficiency: In this approach, the request for critical section access is granted to a process as soon as the token is received. This reduces the waiting time for a process to acquire the critical section.
3. Fairness: The algorithm is fair in the sense that it provides equal opportunity to each process to acquire the critical section. The token is passed sequentially from one process to another, so every process eventually gets a chance to access the critical section.
4. Deadlock-free: The token-based approach ensures that the system remains deadlock-free. A deadlock can only occur if a process that holds a token is not able to access the critical section. However, in this algorithm, the token is always passed to the next process in a sequence, so the system cannot get into a deadlock.

Disadvantages:

1. Single point of failure: In this approach, a central server is responsible for managing the token. If the server fails, the entire system will be affected, and no process will be able to access the critical section.
2. Delay in token acquisition: If a process is waiting for the token, it may take a long time to receive the token. This is especially true for large systems where there are many processes competing for the token.
3. Performance: This approach can be slower than other algorithms because it involves a lot of communication between processes and the central server.
4. Scalability: The token-based approach may not be scalable for large systems with many processes. As the number of processes increases, the token management overhead may become unmanageable.

In summary, the centralized mutual exclusion algorithm with a token-based approach provides simplicity, efficiency, fairness, and deadlock-free execution. However, it also has some drawbacks, such as a single point of failure, delays in token acquisition, performance issues, and scalability problems.

Screenshots of the Output

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated