COURSE NAME: OPERATING SYSTEMS THEORY

COURSE CODE: CS-2006

CONTEXT: PROJECT PROPOSAL

SUBMITTED TO: MS. MUBASHRA FAYYAZ

SUBMITTED BY: 22K-4149 and 22K-4647

SECTION: 4J

PROJECT TITLE: IMPLEMENTATION OF A* SEARCH ALGORITHM USING THREADS

1. **Introduction**:

The A* search algorithm is a widely used path finding and graph traversal algorithm known for its efficiency and optimality. By leveraging threads in its implementation, we aim to enhance its performance and scalability. This project proposes to implement the A* search algorithm using C language and system calls, with a focus on multithreading to exploit parallelism and optimize resource utilization.

2. **Objectives**:

- Implement the A* search algorithm in C language, emphasizing modularity and extensibility.
- Utilize system calls and threading mechanisms to parallelize the algorithm, enhancing its efficiency.
- Develop a comprehensive testing suite to verify the correctness and performance of the implementation.
- Record data pertaining to the efficiency of the threaded A* algorithm, including runtime metrics and resource utilization.

- Analyze the collected data using Python data analysis libraries to derive insights and assess the effectiveness of threading in improving algorithm efficiency.

- Document the implementation process, including design decisions, challenges faced, and lessons learned.

3. Methodology:

- **Algorithm Implementation**: Develop a C implementation of the A* search algorithm, focusing on clarity, efficiency, and adherence to best practices.

- **Threading Integration**: Employ POSIX threads (pthread library) to introduce parallelism into the A* algorithm.

- **Testing and Benchmarking**: Design a comprehensive set of test cases to validate the correctness and performance of the threaded A* implementation. Measure runtime metrics, such as execution time and memory usage, under varying input sizes and complexities.

- **Data Recording**: Implement functionality to record algorithm execution data, including runtime metrics and thread utilization statistics, in a CSV format for further analysis.

- **Data Analysis**: Utilize Python data analysis libraries such as Pandas and Matplotlib to analyze the recorded data. Generate visualizations and statistical summaries to evaluate the impact of threading on algorithm efficiency.

- **Documentation and Reporting**: Maintain detailed documentation throughout the project, including code comments, README files, and a project report summarizing the implementation process, results, and conclusions.

4. Deliverables:

- C implementation of the A* search algorithm with threading support.

- Recorded data in CSV format detailing algorithm efficiency metrics.

- Python scripts for data analysis.

- Well-documented codebase and project report.
- GitHub repository hosting the project code, documentation, and related resources.

5. Tentative Timeline:

- Week 11: Research and design phase, including algorithm understanding and threading concepts.
- Week 12: Implementation of A* algorithm in C, focusing on modularity and scalability.
- Week 12: Integration of threading into the algorithm, ensuring correctness and performance.
- Week 13: Data recording and analysis using Python libraries.
- Week 14: Documentation, report writing, and finalizing project deliverables.

6. Conclusion:

This project aims to explore the effectiveness of threading in optimizing the A* search algorithm's performance. By implementing the algorithm in C language with threading support, recording efficiency metrics, and analyzing the results, we seek to gain insights into the benefits and challenges of parallelizing this widely used search algorithm. Furthermore, the project's findings can serve as a foundation for future research and applications in fields such as AI, gaming, and optimization.

7. GitHub Repository:

Link to GitHub Repository: Click Here