

National University of Computer and Emerging Sciences



<A\* Search Algorithm With Threads>

**Group Members**

<Subhan>…………<22K-4316>

<Shahzaib Ahmed>…………<22K-4390>

<Syed Muhammad Irtiza>…………<22K-4638>

**Supervised by**

**<Miss Anaum Hamid>**

**FAST School of Computing**

**National University of Computer and Emerging Sciences**

**[Main Campus], Pakistan**

**May 2024**

**1. Introduction**

Motivated by the need for faster pathfinding algorithms, our project aims to enhance the efficiency of the A\* search algorithm through the implementation of multithreading. The objective is to parallelize the exploration of paths, leading to quicker results and optimized resource utilization.

**2. Features**

* Utilization of A\* search algorithm: A variation of Dijkstra's Algorithm, A\* search is employed for finding the shortest path between nodes in a weighted graph.
* Multithreading implementation: The project introduces multithreading to explore different paths simultaneously, enhancing the algorithm's speed.
* Heuristic estimation: Manhattan Distance strategy is utilized to estimate the distance between nodes, aiding in efficient pathfinding.

**3. Technology Used**

* Language: C++
* Development Environment: Visual Studio 2019, Dev C++
* Virtualization Platform: Oracle VM VirtualBox (Ubuntu 22.04)

**4.Code Snippets.**

pthread\_t T[8];

pthread\_create(&T[0], NULL, firstSuccessor, NULL);

pthread\_create(&T[1], NULL, secondSuccessor, NULL);

pthread\_create(&T[2], NULL, thirdSuccessor, NULL);

pthread\_create(&T[3], NULL, fourthSuccessor, NULL);

pthread\_create(&T[4], NULL, fifthSuccessor, NULL);

pthread\_create(&T[5], NULL, sixthSuccessor, NULL);

pthread\_create(&T[6], NULL, seventhSuccessor, NULL);

pthread\_create(&T[7], NULL, eighthSuccessor, NULL);

for (int i=0 ; i<8 ; i++) pthread\_join(T[i], NULL);

**5.Output**

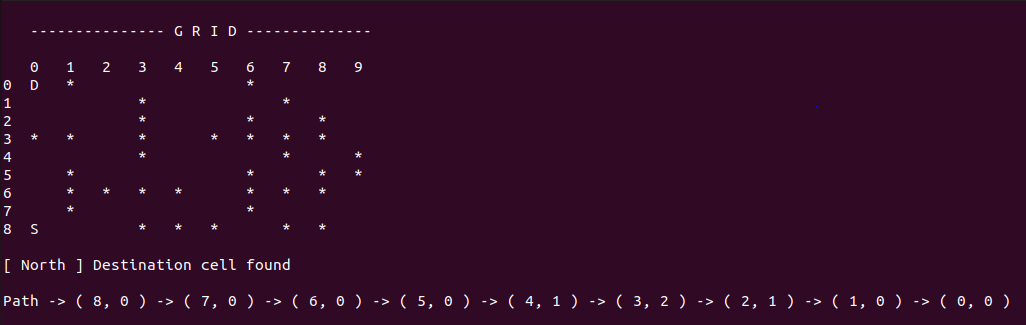
**Windows:**

A picture containing text, screenshot, black

Description automatically generated

**Linux(Ubuntu):**





**6.Conclusion**

In conclusion, the integration of multithreading into the A\* search algorithm has yielded significant improvements in efficiency, allowing for quicker pathfinding across various nodes. However, it's essential to note limitations such as potential thread synchronization issues and increased complexity in code maintenance due to multithreading.

This project serves as a demonstration of how modern computing concepts like multithreading can be applied to traditional algorithms to enhance their performance in real-world applications

**7. References**

* <https://youtu.be/eSOJ3ARN5FM>
* <https://www.geeksforgeeks.org/a-search-algorithm/> (non-multithreading)