Lab 1: Concurrency vs Parallelism

COE892 Rover and Mines

Klajdi Moglica

Note: Registered with AAS for extra time.

Introduction:

This lab focuses on navigating rovers through a 2D minefield and disarming mines using hashing techniques. It involves fetching movement commands, simulating traversal, and implementing mine disarming with both sequential and parallel execution to compare performance and efficiency.

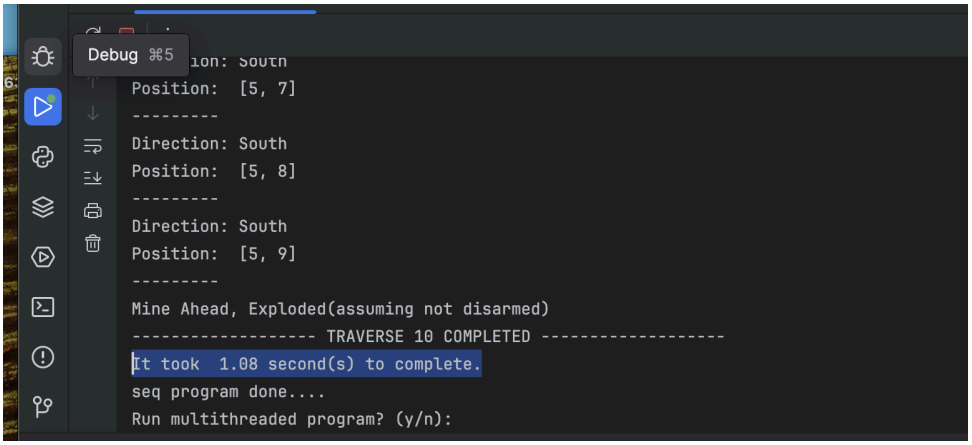
Implementation description:

Part 1 – Simulates moving rovers across a 2D grid (map) and tracks their paths. Each rover executes a series of movement instructions fetched from an API, and the program writes the resulting paths to separate files (path_1.txt, path_2.txt, etc.). This functionality is implemented both sequentially and in parallel.

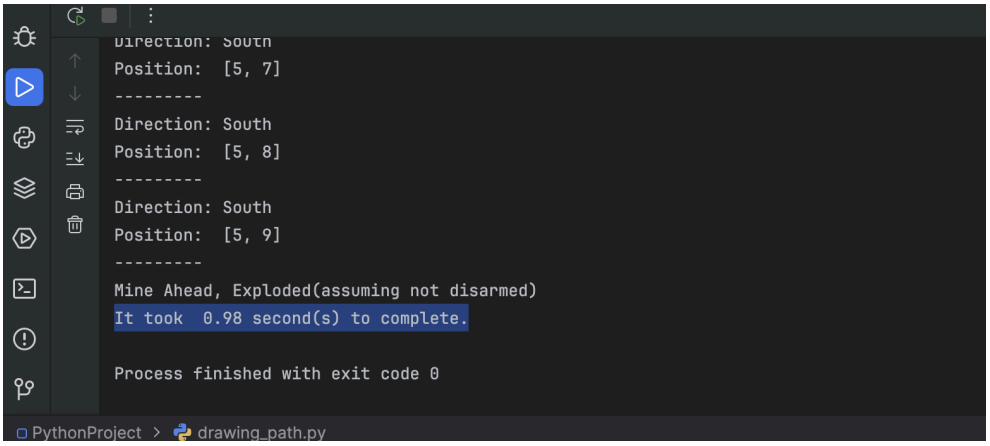
Part 2 – Load the minefield from mines.txt, fetches movement instructions from the fetched API, moves the rover through the minefield, handles mines by using hashing, executes everything first sequentially then in parallel.

Results section:

Part 1 – Drawing path:



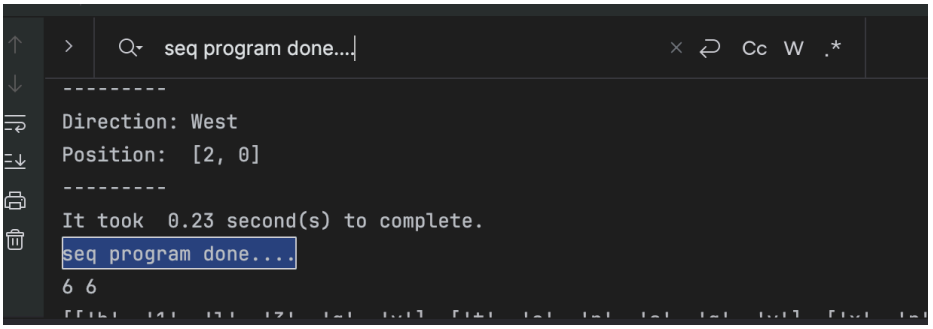
Total processing time: 1.08 seconds for the single threaded program.



Total processing time: 0.98 seconds for the multithreaded program.

Difference of their computation times: The multithreaded program was 0.10 seconds quicker than the single threaded program.

Part 2 – Digging Mines:



Total processing time: 0.23 seconds for the single threaded program.

Total processing time: 0.06 seconds for the multithreaded program.

Difference of their computation times: The multithreaded program was 0.17 seconds quicker than the single threaded program. That's a significant improvement.

Conclusion:

The lab demonstrated effective rover navigation, mine disarming, and performance improvements through multithreading. Sequential and parallel approaches highlighted trade-offs between clarity and efficiency, providing hands-on experience with key programming concepts like file handling, API requests, and multithreading.

I also wanted to mention something interesting I noticed. The third criterion of part 2 adds a computational bottleneck and simulats the real-world effort involved in validating or cracking cryptographic-like hashes. I noticed how long 6 leading zeros were taking, and I realized that finding a valid PIN is not trivial. This part of the lab demonstrated the use of brute-force techniques for solving complex problems.