DINING PHILOSOPHERS' PROBLEM

Mustafa Eren Tuğcu 200316035
Beyza Sungar 200316032
Rukiye Tura 210316049

I. ABSTRACT

The Dining Philosophers problem is a classic challenge in concurrent programming, exploring resource sharing and synchronization. This report presents a detailed analysis of a Python implementation of the problem using the Chandy/Misra algorithm. The code leverages threading, randomness, and visualization through Matplotlib to provide an insightful representation of the concurrent dining experience.

II. INTRODUCTION

The Dining Philosophers problem exemplifies the difficulties in managing shared resources among multiple threads. This analysis delves into the Python implementation, elucidating the underlying Chandy/Misra algorithm and its role in maintaining a synchronized dining process.

III. BACKGROUND

A. CHANDY/MISRA ALGORITHM

The Chandy/Misra algorithm is employed for efficient fork requests and releases among philosophers. It ensures a distributed and deadlock-free solution, critical for avoiding resource conflicts.

B. THREADING AND RANDOMIZATION

Python's threading module is utilized to represent each philosopher as a thread. The incorporation of randomness in thinking and eating durations adds a layer of unpredictability, simulating real-world scenarios.

IV. IMPLEMANTATION

A. FORK CLASS

The Fork class encapsulates the properties of a fork, including an index, lock, ownership status, and methods for acquiring and releasing the fork.

B. PHILOSOPHERS CLASS

The Philosopher class extends the threading. Thread class, representing a philosopher. It manages left and right forks, spaghetti count, and incorporates methods for thinking, requesting forks, eating, and releasing forks.

C. VISUALIZATION

Two visualization methods are implemented:

Animated Table Display: Utilizes Matplotlib to dynamically represent philosopher movements, fork interactions, and spaghetti consumption.

Static Table Output: Prints a concise representation of philosopher and fork states at each step.

V. CONCLUSION

In conclusion, the Python implementation provides a practical understanding of the Dining Philosophers problem. The Chandy/Misra algorithm, coupled with threading and visualization, presents a comprehensive solution to the challenges of resource sharing in concurrent environments.

VI. FUTURE WORK

Future enhancements may include refining the visualization for larger-scale scenarios, exploring alternative synchronization algorithms, and adapting the code for diverse concurrency challenges.

VII. REFERENCES

Chandy, K. M., & Misra, J. (1984). The Drinking Philosophers Problem. ACM Transactions on Programming Languages and Systems, 6(4), 632–646.

VIII. APPENDICES

A. ANIMATED TABLE SCREENSHOTS