MANİSA
CELAL BAYAR
ÜNİVERSİTESİ

PARALLEL PROGRAMING

TERM PROJECT

Emre ÖZCAN

210315072

Sema Nimet ÜNAL

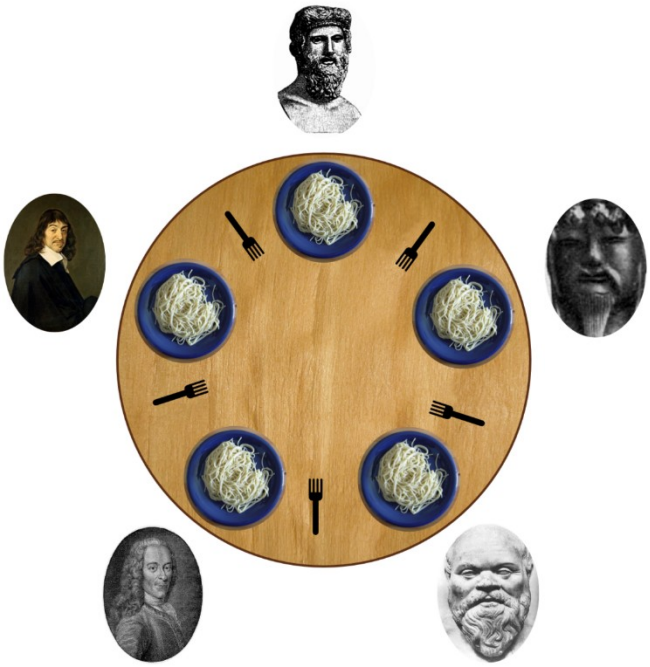
210315095

Elanur İLERİ

200315038

Kerim Batuhan BARAN

200315080



Dijkstra's Solution Strategy for the Dining Philosophers Problem

Dijkstra's solution strategy is an effective method utilized in synchronization-required scenarios like the dining philosophers problem. This strategy aims to minimize hunger situations by establishing a specific order for each philosopher to acquire forks.

The solution was developed using Python's multiprocessing library, following Dijkstra's solution strategy:

- Each philosopher waits in line while attempting to acquire a fork, and observing other philosophers' actions while waiting for their turn.
- Philosophers act based on a semaphores to acquire a fork.
- The queue mechanism ensures a fair distribution of fork sharing. Each philosopher acquires a fork in their order, evenly distributing the duration of hunger.
- After a thinking period, each philosopher expresses a desire to eat.

- Semaphores are used for the fork acquisition process. A philosopher can start eating once they have acquired two forks.
- Philosophers think and eat for random durations.
- After eating, philosophers release the forks and transition to other philosophers.

Key Implementations:

- Semaphore and lock were used to acquire forks.
- The *Philosopher._pick_up_both_forks* function is utilized for acquiring forks, and *Philosopher._put_both_forks_down* for releasing them.
- Shared data is managed using the *multiprocessing.Value* class to facilitate safe data sharing between processes.
- Individual processes are created for each philosopher, running simultaneously.

The shared data is safely managed between processes, ensuring fairness in waiting in line to acquire forks.

The program simulates the thinking and eating processes of the philosophers.