

# Government Engineering College Thrissur

System Software Lab

Navaneeth D

TCR18CS043

S5, CSE

# Dining Philosophers Problem

---

## AIM

Write a program to simulate the working of the dining philosopher's problem.\*

## THEORY

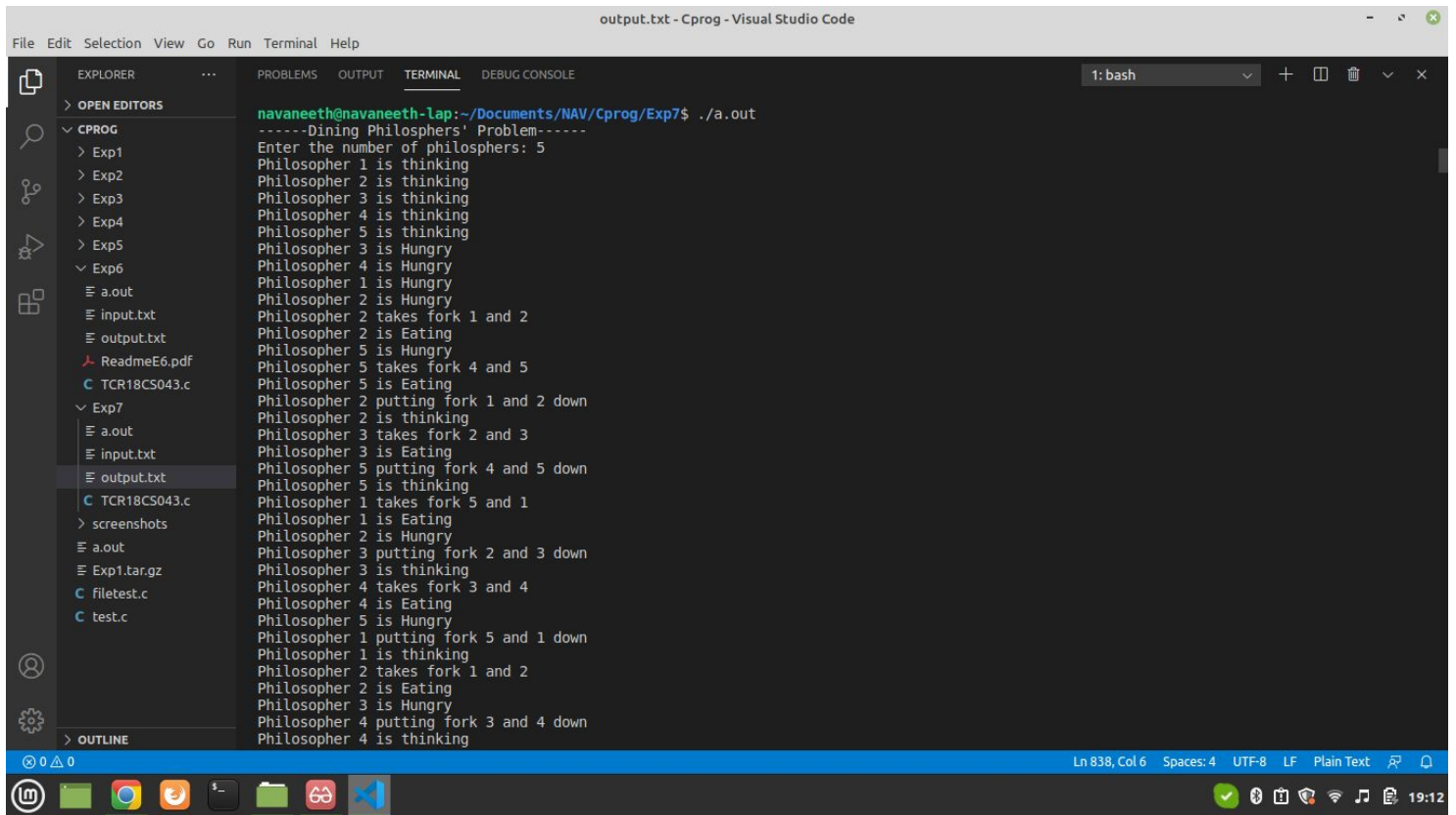
The Dining Philosopher Problem states that K philosophers seated around a circular table with one chopstick between each pair of philosophers. There is one chopstick between each philosopher. A philosopher may eat if he can pick up the two chopsticks adjacent to him. One chopstick may be picked up by any one of its adjacent followers but not both.

There are three states of philosopher : **THINKING, HUNGRY and EATING**. Here there are two semaphores : Mutex and a semaphore array for the philosophers. Mutex is used such that no two philosophers may access the pickup or put down at the same time. The array is used to control the behavior of each philosopher. But, semaphores can result in deadlock due to programming errors.

## RESULT

Dining Philosophers problem was implemented with successful output.

# Output Screenshots



The screenshot shows the Visual Studio Code interface with the terminal window open. The terminal displays the output of a C program titled "Dining Philosophers' Problem". The program prompts the user to enter the number of philosophers, which is 5. The output shows the sequence of actions for 5 philosophers (1 to 5) as they think, become hungry, take forks, eat, and put forks down. The program is running in a terminal window titled "output.txt - Cprog - Visual Studio Code". The Explorer panel on the left shows the project structure with files like "a.out", "input.txt", "output.txt", "ReadmeE6.pdf", "TCR18CS043.c", "screenshots", "a.out", "Exp1.tar.gz", "filetest.c", and "test.c". The status bar at the bottom indicates the current line and column (Ln 838, Col 6) and the file encoding (UTF-8, LF, Plain Text).

```
navaneeth@navaneeth-lap:~/Documents/NAV/Cprog/Exp7$ ./a.out
-----Dining Philosophers' Problem-----
Enter the number of philosophers: 5
Philosopher 1 is thinking
Philosopher 2 is thinking
Philosopher 3 is thinking
Philosopher 4 is thinking
Philosopher 5 is thinking
Philosopher 3 is Hungry
Philosopher 4 is Hungry
Philosopher 1 is Hungry
Philosopher 2 is Hungry
Philosopher 2 takes fork 1 and 2
Philosopher 2 is Eating
Philosopher 5 is Hungry
Philosopher 5 takes fork 4 and 5
Philosopher 5 is Eating
Philosopher 2 putting fork 1 and 2 down
Philosopher 2 is thinking
Philosopher 3 takes fork 2 and 3
Philosopher 3 is Eating
Philosopher 5 putting fork 4 and 5 down
Philosopher 5 is thinking
Philosopher 1 takes fork 5 and 1
Philosopher 1 is Eating
Philosopher 2 is Hungry
Philosopher 3 putting fork 2 and 3 down
Philosopher 3 is thinking
Philosopher 4 takes fork 3 and 4
Philosopher 4 is Eating
Philosopher 5 is Hungry
Philosopher 1 putting fork 5 and 1 down
Philosopher 1 is thinking
Philosopher 2 takes fork 1 and 2
Philosopher 2 is Eating
Philosopher 3 is Hungry
Philosopher 4 putting fork 3 and 4 down
Philosopher 4 is thinking
```