Exp. 10

**Problem Statement:**

Write a program to implement Dining Philosopher’s problem using semaphore **.**

**CODE:-**

**#!/bin/bash**

**# Number of philosophers**

**N=5**

**# Create an array of files to represent forks (semaphores)**

**for ((i = 0; i < N; i++)); do**

**touch "fork\_$i.lock"**

**done**

**# Function for a philosopher**

**philosopher() {**

**local id=$1**

**local left\_fork="fork\_$id.lock"**

**local right\_fork="fork\_$(( (id + 1) % N )).lock"**

**while true; do**

**echo "Philosopher $id is thinking..."**

**sleep $((RANDOM % 3 + 1)) # Simulate thinking time**

**echo "Philosopher $id is hungry..."**

**# Pick up the left fork**

**exec {left\_fd}<>"$left\_fork"**

**flock -x "$left\_fd"**

**# Pick up the right fork**

**exec {right\_fd}<>"$right\_fork"**

**flock -x "$right\_fd"**

**# Eating**

**echo "Philosopher $id is eating..."**

**sleep $((RANDOM % 3 + 1)) # Simulate eating time**

**# Put down the forks**

**flock -u "$right\_fd"**

**flock -u "$left\_fd"**

**echo "Philosopher $id finished eating."**

**sleep $((RANDOM % 3 + 1)) # Simulate time before thinking again**

**done**

**}**

**# Create philosopher processes**

**for ((i = 0; i < N; i++)); do**

**philosopher $i &**

**done**

**# Wait for all philosopher processes (infinite loop simulation)**

**Wait**

**OUTPUT:-**

**A computer screen shot of a computer code

Description automatically generated**