**EXPERIMENT – 12**

12. Design a C program to simulate the concept of Dining-Philosophers problem#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <unistd.h>

#define NUM\_PHILOSOPHERS 5

pthread\_mutex\_t chopsticks[NUM\_PHILOSOPHERS];

void\* philosopher(void\* num) {

int id = \*(int\*)num;

int left = id;

int right = (id + 1) % NUM\_PHILOSOPHERS;

while (1) {

printf("Philosopher %d is thinking...\n", id);

sleep(1);

if (id % 2 == 0) {

pthread\_mutex\_lock(&chopsticks[left]);

pthread\_mutex\_lock(&chopsticks[right]);

} else {

pthread\_mutex\_lock(&chopsticks[right]);

pthread\_mutex\_lock(&chopsticks[left]);

}

printf("Philosopher %d is eating.\n", id);

sleep(2); // Simulate eating

pthread\_mutex\_unlock(&chopsticks[left]);

pthread\_mutex\_unlock(&chopsticks[right]);

printf("Philosopher %d finished eating and starts thinking again.\n", id);

}

return NULL;

}

int main() {

pthread\_t philosophers[NUM\_PHILOSOPHERS];

int ids[NUM\_PHILOSOPHERS];

for (int i = 0; i < NUM\_PHILOSOPHERS; i++) {

pthread\_mutex\_init(&chopsticks[i], NULL);

}

for (int i = 0; i < NUM\_PHILOSOPHERS; i++) {

ids[i] = i;

pthread\_create(&philosophers[i], NULL, philosopher, &ids[i]);

}

for (int i = 0; i < NUM\_PHILOSOPHERS; i++) {

pthread\_join(philosophers[i], NULL);

}

for (int i = 0; i < NUM\_PHILOSOPHERS; i++) {

pthread\_mutex\_destroy(&chopsticks[i]);

}

return 0;

}