**LAB 12 ASSIGNMENT**

**Name:** Birva Babaria

**Roll no.:** CE010

**ID:** 19CEUON064

**1). WAP to implement solution of Sleeping Barber Problem.**

**CODE:**

#include <pthread.h>

#include <semaphore.h>

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

sem\_t customers, barbers, mutex;

int waiting = 0;

const int chairs = 10;

void \*barber()

{

while (1)

{

sem\_wait(&customers);

sem\_wait(&mutex);

waiting--;

sem\_post(&barbers);

sem\_post(&mutex);

puts("barber is cutting");

printf("%d custmoers are in waiting\n", waiting);

sleep(2);

}

}

void \*customer()

{

sem\_wait(&mutex);

if (waiting < chairs)

{

waiting++;

sem\_post(&customers);

sem\_post(&mutex);

sem\_wait(&barbers);

sleep(5);

}

else

{

sem\_post(&mutex);

}

}

int main()

{

pthread\_t barber\_thread, customer\_thread[3];

sem\_init(&customers, 0, 0);

sem\_init(&barbers, 0, 0);

sem\_init(&mutex, 0, 1);

pthread\_create(&barber\_thread, NULL, barber, NULL);

for (int i = 0; i < 10; i++)

{

pthread\_create(&customer\_thread[i], NULL, customer, NULL);

}

pthread\_join(barber\_thread, NULL);

for (int i = 0; i < 10; i++)

{

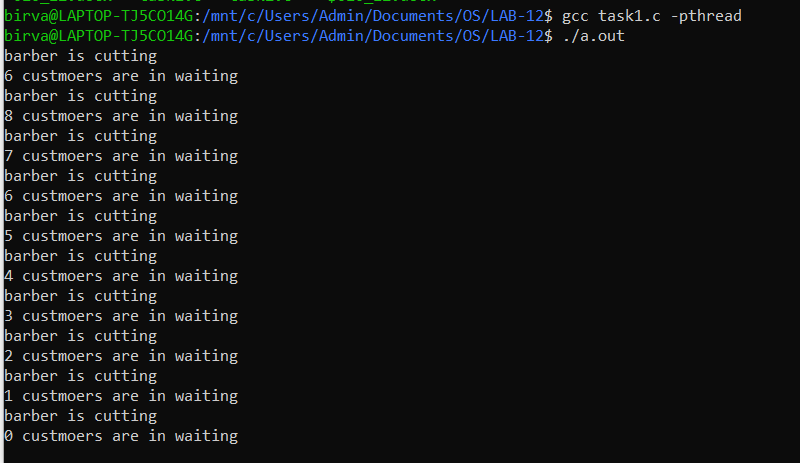
pthread\_join(customer\_thread[i], NULL);

}

return 0;

}

**OUTPUT:**



**2). WAP to implement solution of Dining Philosophers Problem.**

**CODE:**

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <semaphore.h>

#include <unistd.h>

sem\_t chopstick[5];

void \*philos(void \*);

void eat(int);

int main()

{

int i, n[5];

pthread\_t T[5];

for (i = 0; i < 5; i++)

sem\_init(&chopstick[i], 0, 1);

for (i = 0; i < 5; i++)

{

n[i] = i;

pthread\_create(&T[i], NULL, philos, (void \*)&n[i]);

}

for (i = 0; i < 5; i++)

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

}

void \*philos(void \*n)

{

int ph = \*(int \*)n;

printf("Philosopher %d wants to eat\n", ph);

printf("Philosopher %d tries to pick left chopstick\n", ph);

sem\_wait(&chopstick[ph]);

printf("Philosopher %d picks the left chopstick\n", ph);

printf("Philosopher %d tries to pick the right chopstick\n", ph);

sem\_wait(&chopstick[(ph + 1) % 5]);

printf("Philosopher %d picks the right chopstick\n", ph);

eat(ph);

sleep(2);

printf("Philosopher %d has finished eating\n", ph);

sem\_post(&chopstick[(ph + 1) % 5]);

printf("Philosopher %d leaves the right chopstick\n", ph);

sem\_post(&chopstick[ph]);

printf("Philosopher %d leaves the left chopstick\n", ph);

}

void eat(int ph)

{

printf("Philosopher %d begins to eat\n", ph);

}

**OUTPUT:**

