

**Code :**

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <semaphore.h>

#define MAX\_CHAIRS 10

#define CUT\_TIME 1

#define NUM\_BARB 2

#define MAX\_CUST 30

sem\_t customers;

sem\_t barbers;

sem\_t mutex;

int numberOfFreeSeats = MAX\_CHAIRS;

int seatPocket[MAX\_CHAIRS];

int sitHereNext = 0;

int serveMeNext = 0;

static int count = 0;

void barberThread(void \*tmp);

void customerThread(void \*tmp);

void wait();

int main()

{

pthread\_t barber[NUM\_BARB],customer[MAX\_CUST];

int i,status=0;

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

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

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

printf("Good Morning!!\n");

for(i=0;i<NUM\_BARB;i++)

{

status=pthread\_create(&barber[i],NULL,(void )barberThread,(void)&i);

sleep(1);

if(status!=0)

perror("No Barber Present... Sorry!!\n");

}

for(i=0;i<MAX\_CUST;i++)

{

status=pthread\_create(&customer[i],NULL,(void )customerThread,(void)&i);

wait();

if(status!=0)

perror("No Customers Yet!!!\n");

}

for(i=0;i<MAX\_CUST;i++)

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

printf("!!Barber Shop Closes!!\n");

exit(EXIT\_SUCCESS);

}

void customerThread(void \*tmp)

{

int mySeat, B;

sem\_wait(&mutex);

count++;

printf("Customer-%d[Id:%d] Entered Shop. ",count,pthread\_self());

if(numberOfFreeSeats > 0)

{

--numberOfFreeSeats;

printf("Customer-%d Sits In Waiting Room.\n",count);

sitHereNext = (++sitHereNext) % MAX\_CHAIRS;

mySeat = sitHereNext;

seatPocket[mySeat] = count;

sem\_post(&mutex);

sem\_post(&barbers);

sem\_wait(&customers);

sem\_wait(&mutex);

B = seatPocket[mySeat];

numberOfFreeSeats++;

sem\_post(&mutex);

}

else

{

sem\_post(&mutex);

printf("Customer-%d Finds No Seat & Leaves.\n",count);

}

pthread\_exit(0);

}

void barberThread(void \*tmp)

{

int index = \*(int \*)(tmp);

int myNext, C;

printf("Barber-%d[Id:%d] Joins Shop. ",index,pthread\_self());

while(1)

{

printf("Barber-%d Gone To Sleep.\n",index);

sem\_wait(&barbers);

sem\_wait(&mutex);

serveMeNext = (++serveMeNext) % MAX\_CHAIRS;

myNext = serveMeNext;

C = seatPocket[myNext];

seatPocket[myNext] = pthread\_self();

sem\_post(&mutex);

sem\_post(&customers);

printf("Barber-%d Wakes Up & Is Cutting Hair Of Customer-%d.\n",index,C);

sleep(CUT\_TIME);

printf("Barber-%d Finishes. ",index);

}

}

void wait()

{

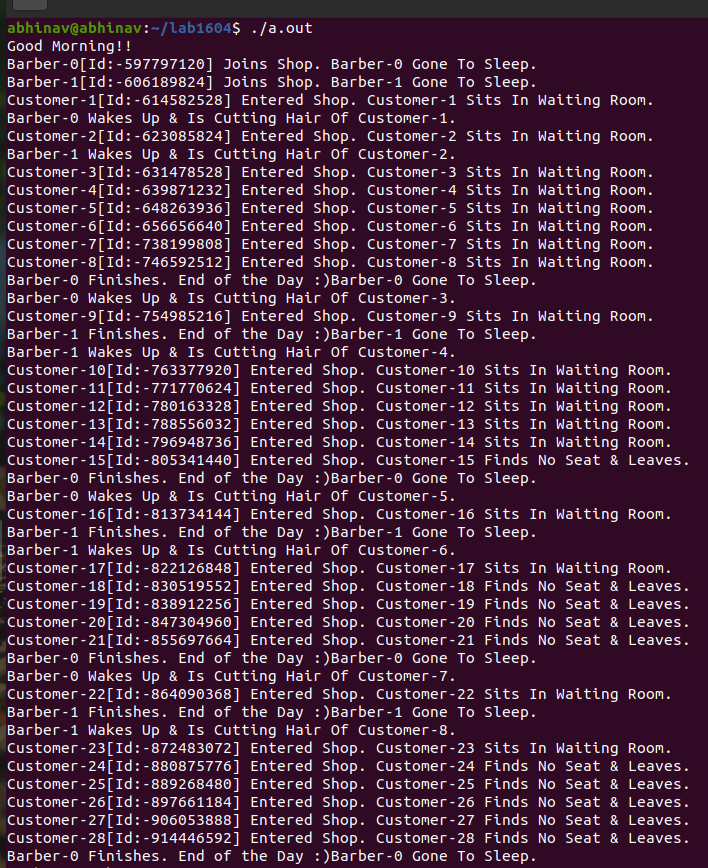
int x = rand() % (250000 - 50000 + 1) + 50000;

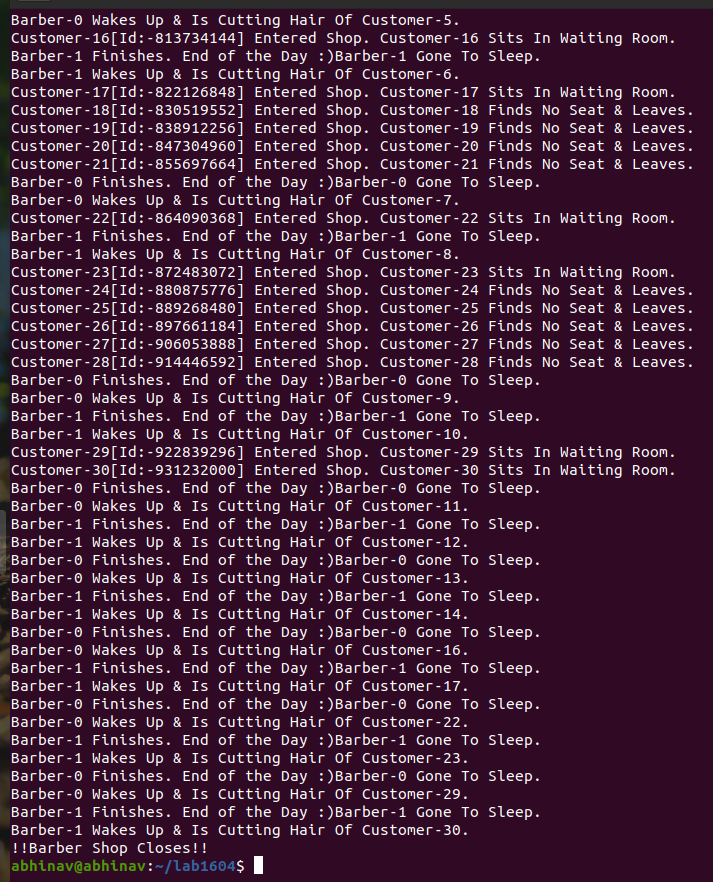
srand(time(NULL));

usleep(x);

}

**Output :**

****

****