NAME: Omkar Jadhav

SECTION: K18GR

ROLL.NO: 65

QUESTION: 13

The problem is to show the relation between customer and barber using the concept of operating system. A barber shop consists of waiting room with n number of chairs and hair cutting room with only 1 chair. When customer enters the shop and if he sees the barber sleeping then he wake up the barber for cutting his hair . If the barber is busy in cutting the hair then customer waits in waiting room and if the the waiting room is full then the customer leaves the shop.

In this problem we will use semaphores to avoid the critical section problem and achieve the process synchronization.

#define MAX\_CUSTOMERS 5

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

#include <time.h>

#include <pthread.h>

#include <semaphore.h>

void \*customer(void \*num);

void \*barber(void \*);

void randwait(int secs);

sem\_t waitingRoom;

sem\_t barberChair;

sem\_t barberPillow;

sem\_t seatBelt;

int allDone = 0;

int main(int argc, char \*argv[])

{

pthread\_t btid;

pthread\_t tid[MAX\_CUSTOMERS];

int i, Customers, Chairs;

int Number[MAX\_CUSTOMERS];

printf("Enter the number of Custmors :");

scanf("%d" , &Customers) ;

printf("Enter the number of Chairs : ");

scanf("%d" , &Chairs);

if (Customers > MAX\_CUSTOMERS)

{

printf("The maximum number of Customers is %d.\n" , MAX\_CUSTOMERS);

exit(-1);

}

for (i=1; i<=MAX\_CUSTOMERS; i++)

{

Number[i] = i;

}

sem\_init(&waitingRoom, 0, Chairs);

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

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

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

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

for (i=1; i<=Customers; i++)

{

pthread\_create(&tid[i], NULL, customer, (void \*)&Number[i]);

sleep(1);

}

for (i=1; i<=Customers; i++)

{

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

sleep(1);

}

allDone = 1;

sem\_post(&barberPillow);

pthread\_join(btid,NULL);

}

void \*customer(void \*number)

{

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

printf("Customer %d arrived at barber shop.\n" , num);

sem\_wait(&waitingRoom);

printf("Customer %d entering waiting room.\n" , num);

sem\_wait(&barberChair);

sem\_post(&waitingRoom);

printf("Customer %d waking the barber.\n" , num);

sem\_post(&barberPillow);

sem\_wait(&seatBelt);

sem\_post(&barberChair);

printf("Customer %d leaving barber shop.\n" , num);

}

void \*barber(void \*junk)

{

while (!allDone)

{

printf("The barber is sleeping\n");

sem\_wait(&barberPillow);

if (!allDone)

{

printf("The barber is cutting hair\n");

randwait(2);

printf("The barber has finished cutting hair.\n");

sem\_post(&seatBelt);

}

}

}

void randwait(int secs)

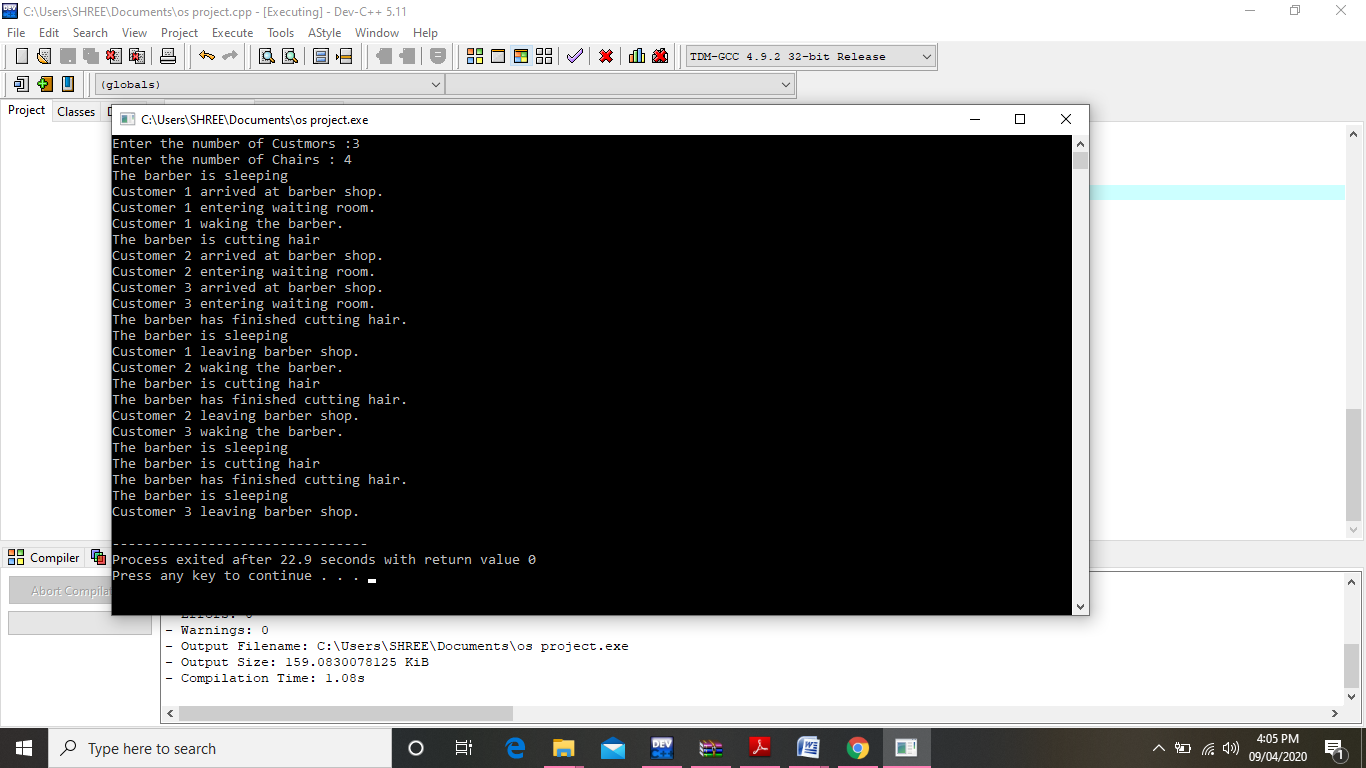
{

int len;

len = (int) ((1 \* secs) + 1);

sleep(len);

}



QUESTION NO.: 16

#include<stdio.h>

#include<conio.h>

#include<unistd.h>

#include<stdlib.h>

int main()

{

int a[10],b[10],x[10];

int waiting[10],turnaround[10],completion[10];

int i,j,s,count=0,t,n;

double avg=0,tt=0,end;

printf("\nEnter the number of Processes: ");

scanf("%d",&n);

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

{

printf("\nEnter the arrival time of process %d : ", i+1);

scanf("%d",&a[i]);

}

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

{

printf("\nEnter the burst time of process %d : ", i+1);

scanf("%d",&b[i]);

}

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

x[i]=b[i];

b[9]=9999;

for(t=0;count!=n;t++)

{

s=9;

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

{

if(a[i]<=t && b[i]<b[s] && b[i]>0 )

s=i;

}

b[s]--;

if(b[s]==0)

{

count++;

end=t+1;

completion[s] = end;

waiting[s] = end - a[s] - x[s];

turnaround[s] = end - a[s];

}

}

printf("pid \t burst \t arrival \twaiting \tturnaround \tcompletion");

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

{

printf("\n %d \t %d \t %d\t\t%d \t\t%d\t\t%d",i+1,x[i],a[i],waiting[i],turnaround[i],completion[i]);

avg = avg + waiting[i];

tt = tt + turnaround[i];

}

printf("\n %If %If",avg,tt);

printf("\n\nThe Average waiting time = %lf\n",avg/n);

printf("The Average Turnaround time = %lf",tt/n);

getch();

}

