LOVELY PROFESSIONAL UNIVERSITY



Simulation Based Assignment

COURSE CODE:-CSE 316

COURSE TITLE:-Operating System

SUBMITTED TO:- Oshin SUBMITTED BY:-

NAME:-Sakshi Kumari Soni

ROLL NO:-34

REGISTRATION NO:-11810588

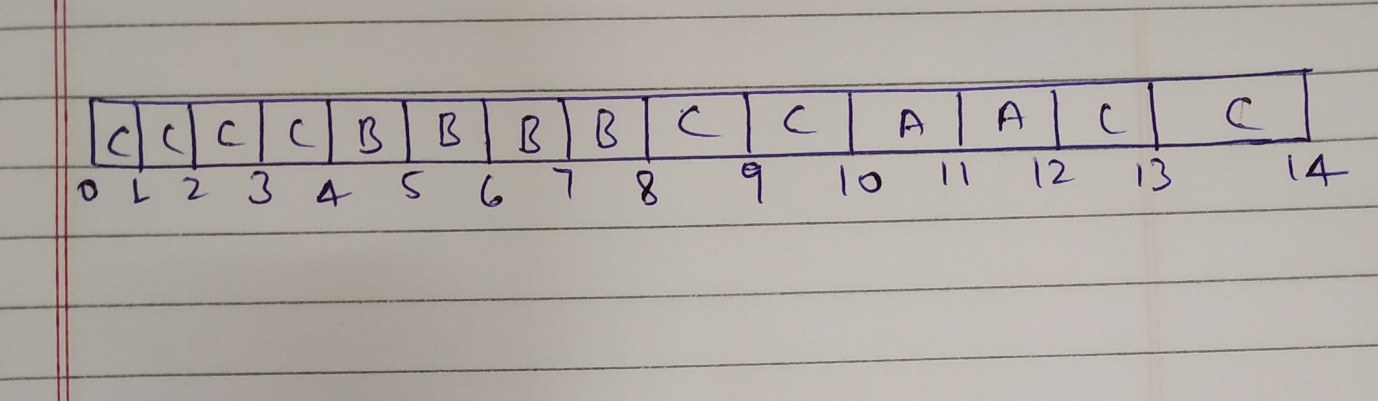
SECTION: K18PT

Q1.Three students(A,B,C) are arriving in the mess at the same time.the id numbers of these students are 2132,2102,2453 and the food taken time from the mess table is 2,4 and 8 minutes.If the two students have same remaining time so it is broken by giving priority to the students with the lowest id number.Consider the longest remaining time first(LRTF) scheduling algorithm and calculate the average turnaround time and waiting time.

Solution:-There are three students A,B and C having id 2132,2102,2453 and they are arrived at same time. Food taken time of students is 2,4,6.This is basically Burst time.We have to apply LRTF(longest remaining time first) algorithm through this we have to find that how food is served between three students.

Because it is LRTF so it execute longest job first, here student C have longest burst time so it will first execute C student.

Gantt Chart:-



Here Student C have maximum 8 burst time so it will execute till reach to 4 then student B and C both have same Burst time 4.but in question mention that if burst time is same then select those student which have Lowest id number. So here student B have lowest id 2102. So now it will execute two time slice then B and student A have same burst time so we have to select a student which have lowest id number so B have lowest id number so it will again execute two times.After that student C have longest burst time than A so it will execute two times then student C and A have same burst time so it will apply lowest id number so A have lowest id number and it will execute two times then we left with only C so it will finish their execution.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S\_id | A.T | B.T | C.T | TAT(CT-AT) | WT(TAT-BT) |
| 2132(A) | 0 | 2 | 12 | 12-0=12 | 12-2=10 |
| 2102(B) | 0 | 4 | 8 | 8-0=8 | 8-4=4 |
| 2453(C) | 0 | 8 | 14 | 14-0=14 | 14-8=6 |

Now we calculate Average TAT=(12+8+14)/3

=11.33 min

Average CT=(10+4+6)/3

=6.67 min

Now Code for this(c language):-

include <stdio.h>

struct student

{

int S\_Id;

int Burst\_time;

int Waiting\_time;

int Turnaround\_time;

};

void get(struct student list[], int t);

void show(struct student list[], int t);

void scheduling(struct student list[], int t);

void waiting\_Time(struct student list[], int x);

void turnAround\_Time(struct student list[], int x);

int main()

{

struct student input[20];

int x,y;

char ch='x';

do

{

printf("Enter how many Students want to eat in mess? : ");

scanf("%d", &x);

get(input,x);

scheduling(input,x);

waiting\_Time(input,x);

turnAround\_Time(input,x);

show(input,x);

printf("Want to proceed further? press 'y' :\n if not then press 'n' ");

scanf("%s",&ch);

}while(ch=='y');

return 0;

}

void get(struct student list[80], int t)

{

int y;

for (y=0;y<t;y++)

{printf("\n\nEnter data for Student #%d",y+1);

printf("\nEnter Student id : ");

scanf("%d",&list[y].S\_Id);

printf("Enter how much time taken for food taking in(min):");

scanf("%d", &list[y].Burst\_time);

}

}

void show(struct student list[80], int t)

{

int y,Avg\_Waiting\_Time=0;

int Avg\_TurnAround\_Time=0;

int Total\_Waiting\_Time=0;

int Total\_TurnAround\_Time=0;

printf("\n\n\t\t\tOutput according to Longest Remaining Time First\n");

printf("\n\t\t\t|===============================================================|");

printf("\n\t\t\t|S\_id\tBurst\_time\tWaiting\_time\tTurnaround\_time |");

printf("\n\t\t\t|===============================================================|");

for (y=0;y<t;y++)

{

printf("\n\t\t\t|%d\t\t%d\t\t%d\t\t%d\t\t|", list[y].S\_Id,list[y].Burst\_time,list[y].Waiting\_time,list[y].Turnaround\_time);

printf("\a\n\t\t\t|---------------------------------------------------------------|");

Total\_Waiting\_Time=Total\_Waiting\_Time+list[y].Waiting\_time;

Total\_TurnAround\_Time=Total\_TurnAround\_Time+list[y].Turnaround\_time;

}

printf("\n\n\t\t\tTotal Waiting Time is: = %d",Total\_Waiting\_Time);

printf("\n\t\t\tTotal Turn around Time is: = %d\n\n",Total\_TurnAround\_Time);

printf("\n\n\t\t\tAverage Waiting Time is: = %d",Total\_Waiting\_Time);

printf("\n\t\t\tAverage Turn around Time is: = %d\n\n",Total\_TurnAround\_Time);

}

void scheduling(struct student list[80], int t)

{

int i, j;

struct student temp;

for(i=0;i<t-1;i++)

{

for(j=0;j<(t - 1-i);j++)

{

if(list[j].Burst\_time<list[j+1].Burst\_time)

{

temp=list[j];

list[j]=list[j+1];

list[j+1]=temp;

}

else if(list[j].Burst\_time==list[j+1].Burst\_time)

{

if(list[j].S\_Id>list[j+1].S\_Id)

{

temp=list[j];

list[j]=list[j+1];

list[j+1]=temp;

}

}

}

}

}

void waiting\_Time(struct student list[80], int x)

{

int p,total;

list[0].Waiting\_time=0;

for(p=1;p<x;p++)

{

list[p].Waiting\_time=list[p-1].Waiting\_time+list[p-1].Burst\_time;

}

}

void turnAround\_Time(struct student list[80], int x)

{

int p,total;

for(p=0;p<x;p++)

{

list[p].Turnaround\_time=list[p].Waiting\_time+list[p].Burst\_time;

}

}

Output of this program:-

