**SSN College of Engineering, Kalavakkam**

**Department of Computer Science and Engineering**

**III Semester - CSE 'A ',’B’ & ‘C’**

**UCS 1312 Data Structures Lab**

**Academic Year: 2019-2020 Batch: 2018-2022**

**Name: Prathyush. S**

**Register Number: 185001112**

**Class: CSE-B**

**Exercise 6: Application of queue**

• Create queue ADT as a header file “queue.h”.

• Create node with two fields namely job number and burst time (jno, bt).

• Assume there are two queues Q1 and Q2.

• Insert the data given below in Q1 and Q2 based on minimum waiting time. (J1, 6), (J2, 5), (J3, 2), (J4, 3), (J5, 7), (J6, 3), (J7, 7), (J8, 2), (J9, 3) and (J10, 7).

• Compute the average waiting time of Q1 and Q2

• Display both the queues with average waiting time

**PROGRAM:-**

**Queue.h:**

#include<stdio.h>

#include<ctype.h>

#include<stdlib.h>

typedef struct Queue{

int front,rear,size,capacity,\*jno,\*bt;

}node;

node \* create(int max){

node \* q;

q=(node\*)malloc(sizeof(node));

q->capacity=max;

q->front=q->size=0;

q->rear=max-1;

q->jno=(int\*)malloc(sizeof(int\*)\*max);

q->bt=(int\*)malloc(sizeof(int\*)\*max);

return q;

}

int avgwait(node\* q){

int avg,sum=0,count=0;

for(int i=0;q->bt[i]!=0;i++){

sum+=q->bt[i];

count++;

}

avg=sum/count;

return avg;

}

int isFull(node \* q){

return(q->capacity==q->size);

}

int isEmpty(node \* q){

return(q->size==0);

}

void enqueue(node \* q, int burst){

if(isFull(q))

return;

q->rear=(q->rear+1)%q->capacity;

q->size++;

q->jno[q->rear]=q->jno[q->rear-1]+1;

q->bt[q->rear]=burst;

printf("\nadded!\n");

}

void dequeue(node \* q){

if(isEmpty(q))

return;

printf("Removing job no: %d and its burst time: %d",q->jno[q->rear],q->bt[q->rear]);

q->jno[q->rear]=0;

q->bt[q->rear]=0;

q->rear=(q->rear-1)%q->capacity;

q->size--;

}

void front(node \* q){

if(isEmpty(q))

printf("\nEmpty");

printf("\nfront...\njob no: %d \t burst time: %d\n",q->jno[q->front],q->bt[q->front]);

}

void rear(node \* q){

if(isEmpty(q))

printf("\nEmpty!");

printf("\nrear....\njob no: %d \t burst time: %d",q->jno[q->rear],q->bt[q->rear]);

}

void view(node \* q){

for(int i=0;q->bt[i]!=0;i++)

printf("\njob no: %d \t burst time: %d",q->jno[i],q->bt[i]);

}

int totwait(node \* q){

int sum=0,count=0;

for(int i=0;q->bt[i]!=0;i++){

sum+=q->bt[i];

count++;

}

return sum;

}

**Main.c:**

#include<stdio.h>

#include “queue.h”

int main()

{

node \*q1,\*q2;

int max,opt=0,n;

printf("Enter total number of jobs: ");

scanf("%d",&max);

q1=create(max);

q2=create(max);

if(!q1 && !q2)

printf("\nFatal error");

if(!q1->jno && !q1->bt)

printf("Error!");

if(!q2->jno && !q2->bt)

printf("Error!");

int bt1,bt2;

printf("\nEnter the burst time for job 1: (seconds) ");

scanf("%d",&bt1);

enqueue(q1,bt1);

printf("\nthis job is assigned to queue: q1");

printf("\nEnter the burst time for job 2: (seconds) ");

scanf("%d",&bt2);

enqueue(q2,bt2);

for(int i=2;i<max;i++){

int bt;

printf("\nEnter the burst time of the job: (seconds) ");

scanf("%d",&bt);

if(totwait(q1)>totwait(q2)){

enqueue(q2,bt);

printf("\nThe job is assigned to q2.");

rear(q2);

}

else if(totwait(q1)<=totwait(q2)){

enqueue(q1,bt);

printf("\nThe job is assigned to q1.");

rear(q1);

}

}

printf("\nthis job is assigned to queue: q2");

do{

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

printf("\nMenu:\n1-veiw average waiting time\n2-view front \n3-view rear \n4-view all \n5-quit\n");

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

printf("\nEnter the required option: ");

scanf("%d",&opt);

if(opt==1){

int avg1,avg2;

avg1=avgwait(q1);

avg2=avgwait(q2);

printf("\nthe average waiting time of q1: %d\nthe average waiting time of q2: %d",avg1,avg2);

}

if(opt==2){

printf("\nQueue : q1...");

front(q1);

printf("\nQueue : q2...");

front(q2);

}

if(opt==3){

printf("\nQueue : q1...");

rear(q1);

printf("\nQueue : q2...");

rear(q2);

}

if(opt==4){

printf("\nQueue : q1...");

view(q1);

printf("\nAverage waiting time: %d",avgwait(q1));

printf("\nQueue : q2...");

view(q2);

printf("\nAverage waiting time: %d",avgwait(q2));

}

}while(opt!=5);

return 0;

}

**OUTPUT:-**

Enter total number of jobs: 10

Enter the burst time for job 1: (seconds) 6

added!

this job is assigned to queue: q1

Enter the burst time for job 2: (seconds) 5

added!

Enter the burst time of the job: (seconds) 2

added!

The job is assigned to q2.

rear....

job no: 2 burst time: 2

Enter the burst time of the job: (seconds) 3

added!

The job is assigned to q1.

rear....

job no: 2 burst time: 3

Enter the burst time of the job: (seconds) 7

added!

The job is assigned to q2.

rear....

job no: 3 burst time: 7

Enter the burst time of the job: (seconds) 3

added!

The job is assigned to q1.

rear....

job no: 3 burst time: 3

Enter the burst time of the job: (seconds) 7

added!

The job is assigned to q1.

rear....

job no: 4 burst time: 7

Enter the burst time of the job: (seconds) 2

added!

The job is assigned to q2.

rear....

job no: 4 burst time: 2

Enter the burst time of the job: (seconds) 3

added!

The job is assigned to q2.

rear....

job no: 5 burst time: 3

Enter the burst time of the job: (seconds) 7

added!

The job is assigned to q1.

rear....

job no: 5 burst time: 7

this job is assigned to queue: q2

======================================

Menu:

1-veiw average waiting time

2-view front

3-view rear

4-view all

5-quit

======================================

Enter the required option: 1

the average waiting time of q1: 5

the average waiting time of q2: 3

======================================

Menu:

1-veiw average waiting time

2-view front

3-view rear

4-view all

5-quit

======================================

Enter the required option: 2

Queue : q1...

front...

job no: 1 burst time: 6

Queue : q2...

front...

job no: 1 burst time: 5

======================================

Menu:

1-veiw average waiting time

2-view front

3-view rear

4-view all

5-quit

======================================

Enter the required option: 3

Queue : q1...

rear....

job no: 5 burst time: 7

Queue : q2...

rear....

job no: 5 burst time: 3

======================================

Menu:

1-veiw average waiting time

2-view front

3-view rear

4-view all

5-quit

======================================

Enter the required option: 4

Queue : q1...

job no: 1 burst time: 6

job no: 2 burst time: 3

job no: 3 burst time: 3

job no: 4 burst time: 7

job no: 5 burst time: 7

Average waiting time: 5

Queue : q2...

job no: 1 burst time: 5

job no: 2 burst time: 2

job no: 3 burst time: 7

job no: 4 burst time: 2

job no: 5 burst time: 3

Average waiting time: 3

======================================

Menu:

1-veiw average waiting time

2-view front

3-view rear

4-view all

5-quit

======================================

Enter the required option: 5