Student Name : Pawan Kumar Singh

Student Id : 11615865

Email Address : [singhpawan9559@gmail.com](mailto:singhpawan9559@gmail.com)

GitHub Link : https://github.com/pawan2880/OS-REPORT

Code : Q.NO-15, Q.NO-18

Q.NO-15

#include<stdio.h>

#include<stdlib.h>

#define PREV\_REQ 125

#define CURR\_REQ 143

int absolute(int num){

return num>0?num:(num\*-1);

}

int compare(const void \* a, const void \* b){

return ( \*(int\*)a - \*(int\*)b );

}

int main(){

int queue[] = {86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130};

int i,s\_index,len=sizeof(queue)/sizeof(int),dist=0,last,curr;

qsort(queue, len, sizeof(int),compare);

last = CURR\_REQ;

//SCAN[ELEVATOR ALGORITHM]

//elevator going up-------->>>>

if(CURR\_REQ>=PREV\_REQ){

for(i=0; i<len; i++){

if(queue[i]>CURR\_REQ){

s\_index=i;

break;

}

}

printf("Order: %4d",last);

for(i=s\_index; i<len; i++){

printf(", %4d",queue[i]);

curr = queue[i];

dist+= absolute(last-curr);

printf("[%d]",absolute(last-curr) );

last = curr;

}

for(i=s\_index; i>0; i--){

printf(", %4d",queue[i-1]);

curr = queue[i];

dist+= absolute(last-curr);

printf("[%d]",absolute(last-curr) );

last = curr;

}

}

//elevator going down----------->>>>>

else if(CURR\_REQ<PREV\_REQ){

for(i=0; i<len; i++){

if(queue[i]>CURR\_REQ){

s\_index=i;

break;

}

}

printf("Order: %4d",last);

for(i=s\_index-1; i>=0; i--){

printf(", %4d",queue[i]);

curr = queue[i];

dist+= absolute(last-curr);

printf("[%d]",absolute(last-curr) );

last = curr;

}

for(i=s\_index; i<len; i++){

printf(", %4d",queue[i]);

curr = queue[i];

dist+= absolute(last-curr);

printf("[%d]",absolute(last-curr) );

last = curr;

}

}

printf("\ntotal distance : %d\n",dist);

return 0\*printf("\nsuccessfully exit\n");

}

TEST CASES-

Q.NO-18

#include<stdio.h>

int main()

{

int p[20],bt[20], su[20], wt[20],tat[20],i, k, n, temp;

float wtavg, tatavg;

printf("\n\*\*\*\*\*\*\*ENTER THE NUMBER OF PROCESS\*\*\*\*\*\*\*\n\n\t\t\t");

scanf("%d",&n);

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

{

p[i] = i; //contains process number

printf("\nENTER THE BURST TIME OF THE PROCESS %d----->>>", i);

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

printf("\nTEACHER/STUDENT PROCESS (0/1) ? ----->>> ");// choose process for teacher or student

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

}

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

{

for(k=i+1;k<n;k++)

{

if(su[i] > su[k])

{

temp=p[i];

p[i]=p[k];

p[k]=temp;

temp=bt[i];

bt[i]=bt[k];

bt[k]=temp;

temp=su[i];

su[i]=su[k];

su[k]=temp;

}

}

}

//calculate wating time

wtavg = wt[0] = 0;

tatavg = tat[0] = bt[0];

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

{

wt[i] = wt[i-1] + bt[i-1];

tat[i] = tat[i-1] + bt[i];

wtavg = wtavg + wt[i]; //calculate average wating time

tatavg = tatavg + tat[i]; //calculate turnaround time

}

printf("\nPROCESS\tTEACHER/STUDENT PROCESS\tBURST TIME\tWAITING TIME\tTURNAROUND TIME");

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

{

printf("\n%d \t\t\ %d \t \t %d \t\t %d \t\t\t %d ",p[i],su[i],bt[i],wt[i],tat[i]);

}

printf("\nAVERAGE WATING IS ----->>> %f",wtavg/n);

printf("\nAVERAGE TURNAROUND TIME ---->>> %f",tatavg/n);

return 0;

}

TEST CASES-