**CPU Scheduling**

#include<iostream>

#include<cmath>

using namespace std;

void fcfs() {

int p[10], bt[10], tt[10], tat[10];

int wt[10], sum, i, no;

int awt, att, at = 0, gc[10];

cout << "\nEnter the no of process:-";

cin >> no;

for (i = 0; i < no; i++) {

cout << "\nEnter burst time of the process p" << i << ":-";

cin >> bt[i];

}

gc[0] = 0;

for (i = 1; i <= no; i++) {

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

}

for (i = 0; i < no; i++) {

wt[i] = gc[i] - at;

}

//Gantt chart

cout << "\n\n Gantt chart:";

cout << "\n-------------------------\n";

for (i = 0; i < no; i++) {

cout << "| p" << i;

}

cout << "|";

cout << "\n--------------------------\n";

for (i = 0; i <= no; i++) {

cout << gc[i] << " ";

}

cout << "\n\nProcess BurstTime WaitingTime TurnAroundTime";

for (i = 0; i < no; i++) {

cout << "\n p" << i << "\t " << bt[i] << "\t\t " << wt[i] << "\t\t " << gc[i + 1];

}

sum = 0;

for (i = 0; i < no; i++) {

sum = sum + wt[i];

}

awt = sum / no;

cout << "\nAverage Waitingn Time=:" << awt;

sum = 0;

for (i = 1; i <= no; i++) {

sum = sum + gc[i];

}

att = sum / no;

cout << "\nAverage Waitingn Time=:" << att;

}

void sjf() {

int bt[10], i, j, wt[10], ft[10], temp, temp\_id, p[10], st[10], tt[10];

int no;

int wat = 0, att = 0;

cout << "\nEnter the no of process:-";

cin >> no;

for (i = 1; i <= no; i++) {

cout << "\nEnter the process id:-";

cin >> p[i];

}

for (i = 1; i <= no; i++) {

cout << "\nEnter burst time of the process :-";

cin >> bt[i];

}

// Sorting

for (i = 1; i <= no; i++) {

for (j = 1; j <= no - 1; j++) {

if (bt[j + 1] < bt[j]) {

temp = bt[j];

bt[j] = bt[j + 1];

bt[j + 1] = temp;

temp\_id = p[j];

p[j] = p[j + 1];

p[j + 1] = temp\_id;

}

}

}

// Gantt chart

cout << "\n\nGantt chart:";

for (i = 1; i <= no; i++) {

cout << "\np" << p[i] << "\t" << bt[i];

}

for (i = 1; i <= no; i++) {

if (i == 1) {

st[i] = 0;

ft[i] = bt[i];

}

else {

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

ft[i] = st[i] + bt[i];

}

}

cout << "\n\n BurstTime WaitingTime TurnAroundTime";

for (i = 1; i <= no; i++) {

wt[i] = st[i];

}

for (i = 1; i <= no; i++) {

tt[i] = ft[i];

}

for (i = 1; i <= no; i++) {

cout << "\n " << bt[i] << "\t\t " << wt[i] << "\t\t " << tt[i];

}

cout << "\n------------------------------------------\n";

// Average waiting time

for (i = 1; i <= no; i++) {

wat = wat + wt[i];

}

wat = wat / no;

cout << "\n\nAverage waiting time:-" << wat;

// Average turn around time

cout << "\n\n Average turn around time :- ";

for (i = 1; i <= no; i++) {

att = att + tt[i];

}

att = att / no;

cout << att;

}

void priority() {

int bt[10], i, j, wt[10], ft[10], temp, temp\_id, p[10], st[10], tt[10];

int no, pr[10], temp\_p;

int wat = 0, att = 0;

cout << "\nEnter the no of process:-";

cin >> no;

for (i = 1; i <= no; i++) {

cout << "\nEnter the process id:-";

cin >> p[i];

}

for (i = 1; i <= no; i++) {

cout << "\nEnter burst time of the process :-";

cin >> bt[i];

}

for (i = 1; i <= no; i++) {

cout << "\nEnter priority of the process:- ";

cin >> pr[i];

}

// Sorting

for (i = 1; i <= no; i++) {

for (j = 1; j <= no - 1; j++) {

if (pr[j + 1] < pr[j]) {

temp\_p = pr[j];

pr[j] = pr[j + 1];

pr[j + 1] = temp\_p;

temp = bt[j];

bt[j] = bt[j + 1];

bt[j + 1] = temp;

temp\_id = p[j];

p[j] = p[j + 1];

p[j + 1] = temp\_id;

}

}

}

// Gantt chart

cout << "\n\nGantt chart:";

for (i = 1; i <= no; i++) {

cout << "\np" << p[i] << "\t" << bt[i] << "\t" << pr[i];

}

for (i = 1; i <= no; i++) {

if (i == 1) {

st[i] = 0;

ft[i] = bt[i];

}

else {

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

ft[i] = st[i] + bt[i];

}

}

cout << "\n\n BurstTime WaitingTime TurnAroundTime";

for (i = 1; i <= no; i++) {

wt[i] = st[i];

}

for (i = 1; i <= no; i++) {

tt[i] = ft[i];

}

for (i = 1; i <= no; i++) {

cout << "\n " << bt[i] << "\t\t" << wt[i] << "\t\t" << tt[i];

}

cout << "\n---------------------------------------------";

// Average waiting time

for (i = 1; i <= no; i++) {

wat = wat + wt[i];

}

wat = wat / no;

cout << "\n\nAverage waiting time:-" << wat;

// Average turn around time

cout << "\n\n Average turn around time :- ";

for (i = 1; i <= no; i++) {

att = att + tt[i];

}

att = att / no;

cout << att;

}

void round\_robin() {

int arr[20], awt[10], no, ch;

int tat[30], pro[30], rr[30], pro1[30], bt[30], flag[15],

flag1[20], gc[20];

int i, j, k, at, sum, tq, rnd\_up, a = 0;

float avg, temp;

at = sum = temp = 0;

cout << "\n\nEnter the number of processes you want: ";

cin >> no;

cout << "\nEnter the time quantum: ";

cin >> tq;

//------------Enter process burst time-----------

for (i = 0; i < no; i++) {

cout << "\nEnter the process number, burst time: ";

cin >> pro[i] >> arr[i];

flag[i] = flag1[i] = gc[i] = 0;

bt[i] = arr[i]; // store the burst time

if (temp <= arr[i])

temp = arr[i];

}

temp = temp / tq;

rnd\_up = ceil(temp); // No of Passes

k = 0;

rr[k] = 0;

for (i = 0; i < rnd\_up; i++) // No of Passes

{

for (j = 0; j < no; j++) // No of Processes

{

if (arr[j] != 0) // Check if arr[j] != 0

{

if (arr[j] >= tq) {

rr[++k] = tq;

pro1[k - 1] = pro[j];

arr[j] = arr[j] - tq;

}

else {

rr[++k] = arr[j];

pro1[k - 1] = pro[j];

arr[j] = 0;

}

}

}

}

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

arr[i] = arr[i - 1] + rr[i];

}

cout << "\n\tGANTT CHART";

cout << "\n-----------------------------------------------\n";

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

cout << "| P" << pro1[i];

}

cout << "|";

cout << "\n-----------------------------------------------\n";

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

cout << arr[i] << " ";

//----------------Waiting Time------------

cout << "\n-------------------------------\n";

for (i = 0; i < no; i++) {

a = i; // set a to ith position

gc[i] = gc[i] + arr[a];

for (j = i + 1; j < k; j++) {

if (pro1[j] == pro1[i]) {

gc[i] = gc[i] + arr[j] - arr[a + 1];

a = j;

flag[i] = 1;

}

}

tat[i] = arr[a + 1];

a = 0;

}

//-------------------Printing--------------

for (i = 0; i < no; i++) {

if (flag[i] == 0) {

gc[i] = arr[i];

}

}

cout << "\n\nProcess\tBurst time\tWaiting time\tTurn Around Time";

for (i = 0; i < no; i++) {

cout << "\nP" << pro[i] << "\t" << bt[i] << "\t\t" << gc[i] << "\t\t" << tat[i];

}

//---------Average Waiting Time-------

sum = 0;

for (i = 0; i < no; i++) {

sum = sum + gc[i];

}

avg = sum / no;

cout << "\n\nAverage Waiting Time: " << avg;

//--------Average Turn Around Time-------

sum = 0;

for (i = 0; i < no; i++) {

sum = sum + tat[i];

}

avg = sum / no;

cout << "\n\nThe Average Turn Around Time: " << avg;

}

int main() {

int ch;

char ans;

do {

cout << "\n\*\*\*\*MENU\*\*\*\*\*\* ";

cout << "\n1.FCFS";

cout << "\n2.SJF";

cout << "\n3.PRIORITY";

cout << "\n4.ROUND ROBIN";

cout << "\n5.Exit";

cout << "\nEnter your choice:- ";

cin >> ch;

switch (ch) {

case 1:

fcfs();

break;

case 2:

sjf();

break;

case 3:

priority();

break;

case 4:

round\_robin();

break;

case 5:

return 0; // Using 'return 0' instead of 'exit()'

default:

break;

}

cout << "\n\nDo you want to continue? (y/n): ";

cin >> ans;

} while (ans == 'y' || ans == 'Y');

return 0;

}









