Lab 5

1. Write a menu driven program where the user gets to choose his/her choice of scheduling method.

Code:

#include <iostream>

using namespace std;

struct Process {

int pid;

int bt;

int art;

};

void fcfs() {

int n, bt[20], wt[20], tat[20], i, j;

float avwt = 0, avtat = 0;

cout << "Enter number of processes:";

cin >> n;

cout << "\nEnter Burst Time \n";

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

cout << "Burst time[" << i + 1 << "]:";

cin >> bt[i];

}

wt[0] = 0;

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

wt[i] = 0;

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

wt[i] += bt[j];

}

cout << "\nProcess\t\tBurst Time\tWaiting Time\tTurnaround Time";

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

tat[i] = bt[i] + wt[i];

avwt += wt[i];

avtat += tat[i];

cout << "\nP[" << i + 1 << "]\t\t\t" << bt[i] << "\t\t\t" << wt[i] << "\t\t\t\t" << tat[i];

}

avwt = avwt / n;

avtat = avtat / n;

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

cout << "\nAverage Turnaround Time:" << avtat;

}

void sjf() {

int A[100][4];

int i, j, n, total = 0, index, temp;

float avg\_wt, avg\_tat;

cout << "Enter number of processes: ";

cin >> n;

cout << "Enter Burst Time:" << endl;

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

cout << "P" << i + 1 << ": ";

cin >> A[i][1];

A[i][0] = i + 1;

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

index = i;

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

if (A[j][1] < A[index][1])

index = j;

temp = A[i][1];

A[i][1] = A[index][1];

A[index][1] = temp;

temp = A[i][0];

A[i][0] = A[index][0];

A[index][0] = temp;

}

A[0][2] = 0;

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

A[i][2] = 0;

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

A[i][2] += A[j][1];

total += A[i][2];

}

avg\_wt = (float)total / n;

total = 0;

cout << "P\tBT\tWT\tTAT" << endl;

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

A[i][3] = A[i][1] + A[i][2];

total += A[i][3];

cout << "P" << A[i][0] << "\t" << A[i][1] << "\t" << A[i][2] << "\t" << A[i][3] << endl;

}

avg\_tat = (float)total / n;

cout << "Average Waiting Time= " << avg\_wt << endl;

cout << "Average Turnaround Time= " << avg\_tat << endl;

}

void roundRobin() {

int i, n, time, remain, temps = 0, time\_quantum;

int wt = 0, tat = 0;

cout << "Enter the total number of processes: ";

cin >> n;

remain = n;

int at[n], bt[n], rt[n];

cout << "Enter the Arrival time, Burst time for All the processes" << endl;

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

cout << "Arrival time for process " << i + 1 << ": ";

cin >> at[i];

cout << "Burst time for process " << i + 1 << ": ";

cin >> bt[i];

rt[i] = bt[i];

}

cout << "Enter the value of time QUANTUM: ";

cin >> time\_quantum;

cout << "\n\nProcess\tTurnaround Time\tWaiting Time\n\n";

for (time = 0, i = 0; remain != 0;) { if (rt[i] <= time\_quantum && rt[i] > 0) {

time += rt[i];

rt[i] = 0;

temps = 1;

} else if (rt[i] > 0) {

rt[i] -= time\_quantum;

time += time\_quantum;

}

if (rt[i] == 0 && temps == 1) {

remain--;

cout << "Process " << i + 1 << "\t" << time - at[i] << "\t\t" << time - at[i] - bt[i] << endl;

wt += time - at[i] - bt[i];

tat += time - at[i];

temps = 0;

}

if (i == n - 1)

i = 0;

else if (at[i + 1] <= time)

i++;

else

i = 0;

}

cout << "Average waiting time " << wt \* 1.0 / n << endl;

cout << "Average turn around time " << tat \* 1.0 / n << endl;

}

void srtn() {

int n;

cout << "Enter number of processes: ";

cin >> n;

Process \*proc = new Process[n];

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

cout << "Enter details for Process " << i + 1 << ":" << endl;

cout << "Burst Time: ";

cin >> proc[i].bt;

cout << "Arrival Time: ";

cin >> proc[i].art;

proc[i].pid = i + 1;

}

int wt[n] = {0}, tat[n] = {0}, total\_wt = 0, total\_tat = 0;

int rt[n];

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

rt[i] = proc[i].bt;

}

int complete = 0, t = 0;

bool check = false;

while (complete != n) {

int shortest = -1;

for (int j = 0; j < n; j++) {

if (proc[j].art <= t && rt[j] > 0) {

if (shortest == -1 || rt[j] < rt[shortest]) {

shortest = j; }

check = true;

}

}

if (!check) {

t++;

continue;

}

rt[shortest]--;

if (rt[shortest] == 0) {

complete++;

check = false;

int finish\_time = t + 1;

wt[shortest] = finish\_time - proc[shortest].bt - proc[shortest].art;

if (wt[shortest] < 0) {

wt[shortest] = 0;

}

}

t++;

}

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

tat[i] = proc[i].bt + wt[i];

}

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

total\_wt += wt[i];

total\_tat += tat[i];

}

cout << "Processes Burst time Waiting time Turnaround time\n";

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

cout << " " << proc[i].pid << "\t\t" << proc[i].bt << "\t\t " << wt[i] << "\t\t " << tat[i] << endl;

}

cout << "\nAverage waiting time = " << (float)total\_wt / n;

cout << "\nAverage turn around time = " << (float)total\_tat / n << endl;

delete[] proc;

}

int main() {

int choice;

do {

cout << "\nMenu:\n";

cout << "1. FCFS Scheduling\n";

cout << "2. SJF Scheduling\n";

cout << "3. Round Robin Scheduling\n";

cout << "4. SRTN Scheduling\n";

cout << "5. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

fcfs();

break;

case 2:

sjf();

break;

case 3:

roundRobin(); break;

case 4:

srtn();

break;

case 5:

cout << "Exiting program...\n";

break;

default:

cout << "Invalid choice! Please enter again.\n";

}

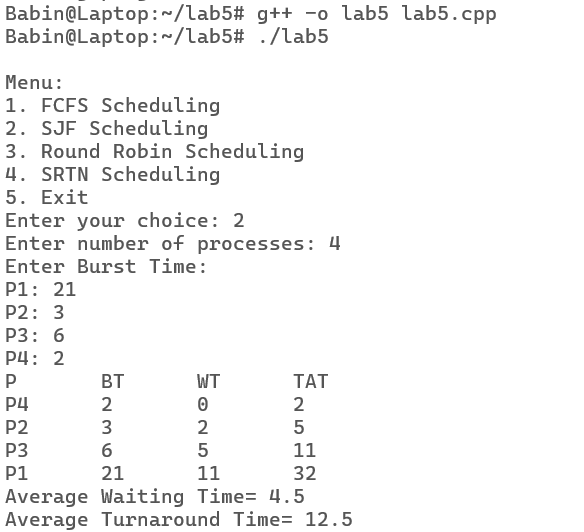
} while (choice != 5);

return 0;

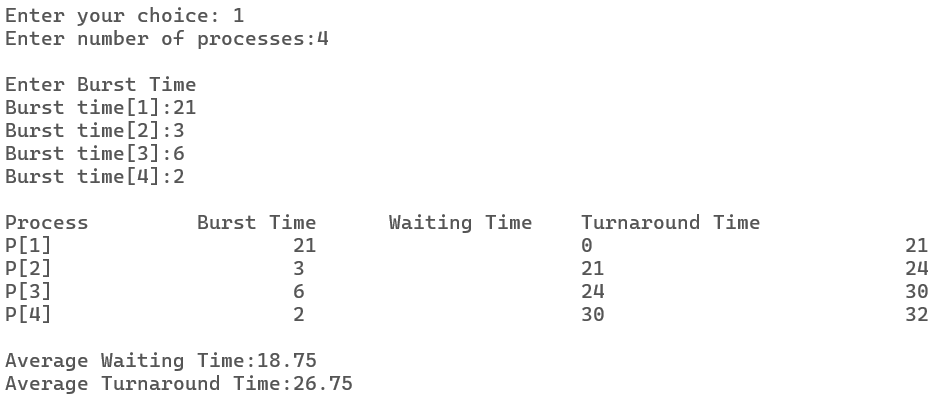
}

OUTPUT:

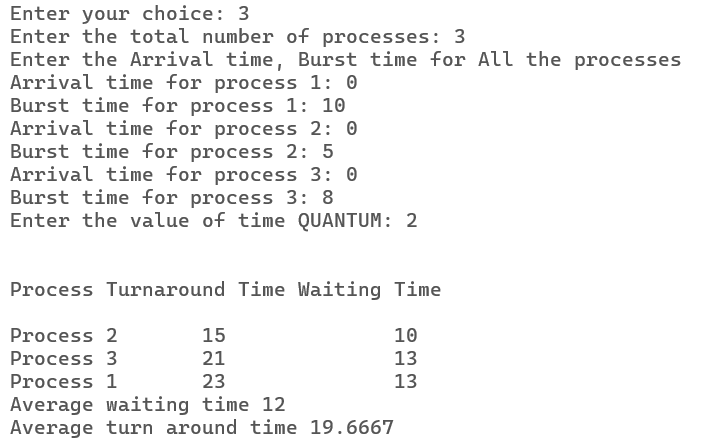
For SJF:



For First Come First Serve:



For Round Robin:



For Shortest Time Remaining Next:

