**Lab:1**

**FCFS scheduling in C**

**Code:**

#include<stdio.h>

int main(){

int process[10], i, n, wt[10], tt[10], aw = 0, at = 0;

printf("Enter number of processes: ");

scanf("%d", &n);

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

printf("Enter the burst time of process %d: ", i + 1);

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

}

wt[0] = 0;

printf("Process\t\tBurst Time\t\tWaiting Time\tTurn Around Time\n");

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

tt[i] = wt[i] + process[i];

printf("P[%d]\t\t\t%d\t\t\t\t%d\t\t\t\t%d\n", i + 1, process[i], wt[i], tt[i]);

wt[i + 1] = wt[i] + process[i];

}

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

aw = aw + wt[i];

at = at + tt[i];

}

aw = aw / n;

at = at / n;

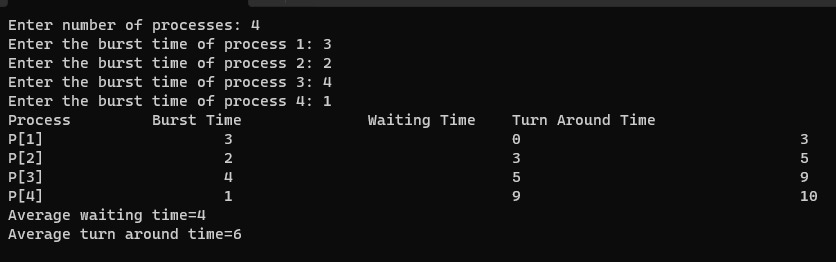
printf("Average waiting time=%d\n", aw);

printf("Average turn around time=%d\n", at);

return 0;

}

Output:



**Lab 2:**

**Shortest Job first using C**

**Code:**

**#include <stdio.h>**

int main() {

int at[10], bt[10], i, n, t = 0, count = 0, att[10], btt[10], process[10], j = 0, tat[10], wt[10], bt2[10];

float avgtat = 0, avgwt = 0;

printf("Enter number of processes: ");

scanf("%d", &n);

printf("Enter the arrival time and burst time:\n");

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

printf("P%d:", i + 1);

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

bt2[i] = bt[i];

}

printf("Process\t Arrival\_Time\tBurst\_Time\tTurn\_Around\_Time\tWaiting\_Time\n");

while (count < n) {

int count1 = 0;

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

if (at[i] <= t && bt[i] != 0) {

att[count1] = at[i];

btt[count1] = bt2[i];

process[count1] = i + 1;

count1++;

}

}

for (i = 0; i < count1 - 1; i++) {

for (j = i + 1; j < count1; j++) {

int temp;

if (btt[i] > btt[j]) {

temp = btt[i];

btt[i] = btt[j];

btt[j] = temp;

temp = att[i];

att[i] = att[j];

att[j] = temp;

temp = process[i];

process[i] = process[j];

process[j] = temp;

}

}

}

t++;

bt[process[0] - 1]--;

if (bt[process[0] - 1] == 0) {

tat[process[0] - 1] = t - att[0];

wt[process[0] - 1] = tat[process[0] - 1] - btt[0];

avgtat += tat[process[0] - 1];

avgwt += wt[process[0] - 1];

printf("P%d\t\t%d\t\t%d\t\t%d\t\t\t%d\n", process[0], att[0], btt[0], tat[process[0] - 1], wt[process[0] - 1]);

count++;

}

}

avgtat = avgtat / n;

avgwt = avgwt / n;

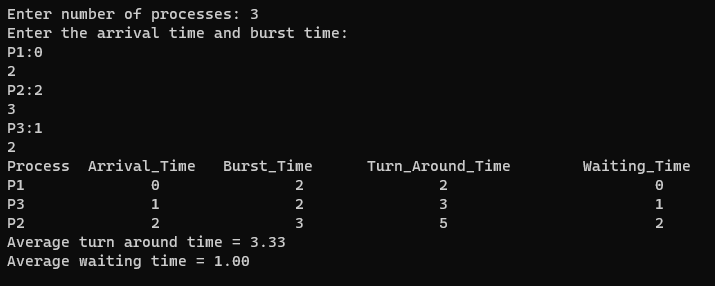
printf("Average turn around time = %.2f\n", avgtat);

printf("Average waiting time = %.2f\n", avgwt);

return 0;

}

Output:



**Lab 3:**

**Priority scheduling non preemptive using c**

**Code:**

#include<stdio.h>

int main(){

int at[10], bt[10], pr[10], i, n, t = 0, count = 0, att[10], btt[10], prr[10], process[10], j = 0, count1 = 0, tat[10], wt[10];

float avgtat = 0, avgwt = 0;

printf("Enter number of processes: ");

scanf("%d", &n);

printf("Enter the arrival time, burst time, and priority of\n");

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

printf("P%d:", i + 1);

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

}

printf("Process\t Arrival\_Time\tBurst\_Time\tPriority\tTurn\_Around\_Time\tWaiting\_Time\n");

while (count1 < n) {

count = 0;

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

if (at[i] <= t && bt[i] != 0) {

att[count] = at[i];

btt[count] = bt[i];

prr[count] = pr[i];

process[count] = i + 1;

count++;

}

}

for (i = 0; i < count - 1; i++) {

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

int temp;

if (prr[i] > prr[j]) {

temp = prr[i];

prr[i] = prr[j];

prr[j] = temp;

temp = btt[i];

btt[i] = btt[j];

btt[j] = temp;

temp = att[i];

att[i] = att[j];

att[j] = temp;

temp = process[i];

process[i] = process[j];

process[j] = temp;

}

}

}

t = t + btt[0];

tat[process[0] - 1] = t - att[0];

wt[process[0] - 1] = tat[process[0] - 1] - btt[0];

bt[process[0] - 1] = 0;

count1++;

avgtat = avgtat + tat[process[0] - 1];

avgwt = avgwt + wt[process[0] - 1];

printf("P%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t\t%d\n", process[0], att[0], btt[0], prr[0], tat[process[0] - 1], wt[process[0] - 1]);

}

avgtat = avgtat / n;

avgwt = avgwt / n;

printf("Average turn around time = %f\n", avgtat);

printf("Average waiting time = %f\n", avgwt);

return 0;

}

Output:

