National University of Modern Languages

Lab Report#02

Roll # 2340

Class: BSCS 5B Morning

Subject: Operating System(Lab)

Submitted to: Mrs. Humaira Batool

Submitted by: Farhan Haider

FCFS(With Arrival Time):

```
// C++ program for implementation of FCFS
// scheduling with different arrival time
#include<iostream>
using namespace std;

// Function to find the waiting time for all
// processes
void findWaitingTime(int processes[], int n, int bt[],int wt[], int at[])
{
        int service_time[n];
        service_time[0] = at[0];
        wt[0] = 0;

        // calculating waiting time
        for (int i = 1; i < n ; i++)
        {
                // Add burst time of previous processes
                service_time[i] = service_time[i-1] + bt[i-1];

                // Find waiting time for current process =
                // sum - at[i]
                wt[i] = service_time[i] - at[i];

                // If waiting time for a process is in negative
        // that means it is already in the ready queue
        // before CPU becomes idle so its waiting time is 0
        if (wt[i] < 0)
            wt[i] = 0;
    }
}

// Function to calculate turn around time
void findTurnAroundTime(int processes[], int n, int bt[],int wt[], int tat[])
{
    // Calculating turnaround time by adding bt[i] + wt[i]
    for (int i = 0; i < n ; i++)
        tat[i] = bt[i] + wt[i];
}

// Function to calculate average waiting and turn-around
// times.
void findavgTime(int processes[], int n, int bt[], int at[])
{
    int wt[n], tat[n];

    // Function to find waiting time of all processes
    findWaitingTime(processes, n, bt, wt, at);

    // Function to find turn around time for all processes
    findTurnAroundTime(processes, n, bt, wt, tat);


    // Display processes along with all details
    cout << "Processes " << " Arrival Time " << " Burst Time "<< " Completion
Time " << " Turn-Around Time "<< " waiting Time \n";
    int total_wt = 0, total_tat = 0;
    for (int i = 0 ; i < n ; i++)
    {
        total_wt = total_wt + wt[i];
        total_tat = total_tat + tat[i];
        int compl_time = tat[i] + at[i];
        cout << "P" << i+1 << "\t\t" << at[i] << "\t\t"<< bt[i] << "\t\t" <<
compl_time << "\t\t "<< tat[i] << "\t\t " << wt[i]  << endl;
    }


    cout << "Average waiting time = "<< (float)total_wt / (float)n;
    cout << "\nAverage turn around time = "<< (float)total_tat / (float)n;
}


// Driver code
int main()
{
    // Process id's
    int processes[] = {1, 2, 3, 4, 5};
    int n = sizeof processes / sizeof processes[0];
```

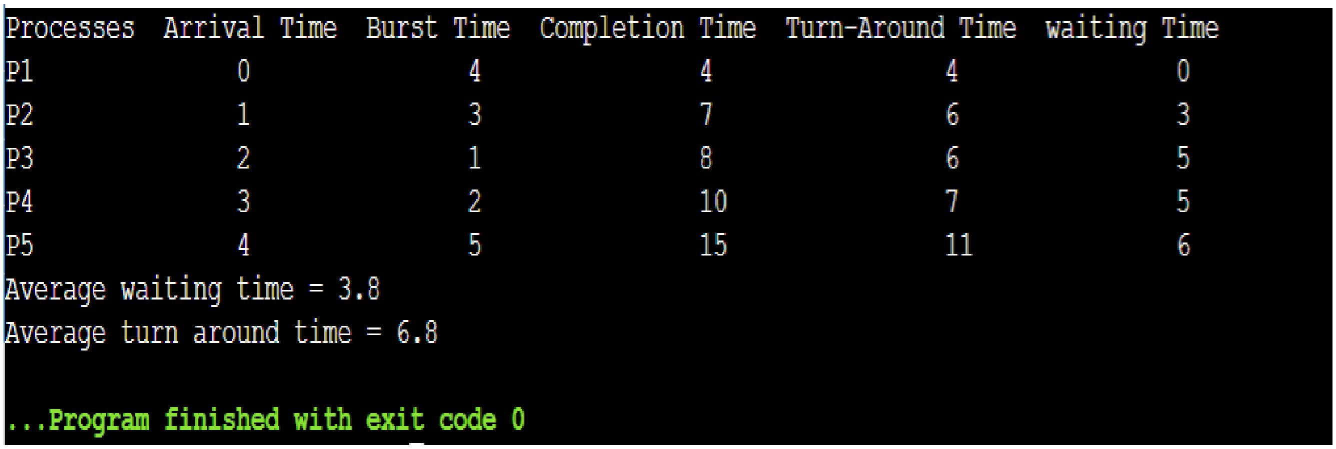
```
	// Burst time of all processes
	int burst_time[] = {4,3,1,2,5};

	// Arrival time of all processes
	int arrival_time[] = {0,1,2,3,4};

	findavgTime(processes, n, burst_time, arrival_time);

	return 0;
}
```

Output:



```
Processes  Arrival Time  Burst Time  Completion Time  Turn-Around Time  waiting Time
P1              0             4             4                4               0
P2              1             3             7                6               3
P3              2             1             8                6               5
P4              3             2             10               7               5
P5              4             5             15               11              6
Average waiting time = 3.8
Average turn around time = 6.8

...Program finished with exit code 0
```