**National University of Modern Languages**

Logo

Description automatically generated

**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:**

A screenshot of a computer

Description automatically generated with medium confidence