Lab No. 4 **Date: 2025/09/01**

Title: Write a program to calculate the average turnaround time and waiting time for user input process parameters using FCFS process scheduling algorithm.

Introduction

First Come First Serve (FCFS) is the simplest CPU scheduling algorithm.

- The process that arrives first is executed first.
- It is a non-preemptive algorithm (once a process starts, it cannot be interrupted).
- It follows the FIFO (First In First Out) principle.

Terminologies

- Burst Time (BT): Time required by a process for execution.
- Arrival Time (AT): Time when a process arrives in the ready queue. (Here we assume AT = 0 for all processes).
- Completion Time (CT): Time when a process finishes execution.
- Turnaround Time (TAT): $TAT = CT - AT$
- Waiting Time (WT): $WT = TAT - BT$

Advantages

- Simple and easy to implement.
- Fair for processes (executed in arrival order).

Disadvantages

- Convoy effect: Long processes delay shorter ones.
- Poor performance for varying burst times.

Programming Language: C++

IDE: Dev C++

Source Code:

```
#include <iostream>
#include <iomanip>
using namespace std;
struct Process
{
   string name;
   int arrival;
   int burst;
   int completion;
   int turnaround;
   int waiting;
};

void sortByArrival(Process p[], int n)
{
   for (int i = 0; i < n - 1; i++)
   {
      for (int j = 0; j < n - i - 1; j++)
      {
         if (p[j].arrival > p[j + 1].arrival)
         {
            swap(p[j], p[j + 1]);
         }
      }
   }
}

int main()
{
   cout << "====================================" << endl;
   cout << "||      FCFS Scheduling        ||" << endl;
   cout << "|| Compiled by :- Jonash Chataut  ||" << endl;
   cout << "====================================" << endl << endl;
   int n;
   cout << "Enter number of processes: ";
   cin >> n;
   Process p[n];
   cout << "Enter process name, arrival time, and burst time:\n";
   for (int i = 0; i < n; i++)
   {
      cin >> p[i].name >> p[i].arrival >> p[i].burst;
   }
   sortByArrival(p, n);
   int currentTime = 0;
   double totalTAT = 0, totalWT = 0;
   // Calculate CT, TAT, WT
   for (int i = 0; i < n; i++)
   {
```

```
        if (p[i].arrival > currentTime)
        {
            currentTime = p[i].arrival; // CPU idle
        }
        p[i].completion = currentTime + p[i].burst;
        currentTime = p[i].completion;
        p[i].turnaround = p[i].completion - p[i].arrival; // TAT = CT - AT
        p[i].waiting = p[i].turnaround - p[i].burst;     // WT = TAT - BT
        totalTAT += p[i].turnaround;
        totalWT += p[i].waiting;
    }
    // Print Gantt Chart
    cout << "\nGantt Chart:\n";
    cout << "-------------------------------------------------\n";
    for (int i = 0; i < n; i++)
    {
        cout << "|   " << p[i].name << "   ";
    }
    cout << "|\n";
    cout << "-------------------------------------------------\n";
    cout << p[0].arrival;
    for (int i = 0; i < n; i++)
    {
        cout << setw(8) << p[i].completion;
    }
    cout << "\n";
    // Print Process Table
    cout << "\nProcess\tAT\tBT\tCT\tTAT (CT-AT)\tWT (TAT-BT)\n";
    cout << "-----------------------------------------------------------------\n";
    for (int i = 0; i < n; i++)
    {
        cout << p[i].name << "\t"
             << p[i].arrival << "\t"
             << p[i].burst << "\t"
             << p[i].completion << "\t"
             << p[i].completion << " - " << p[i].arrival << " = " << p[i].turnaround << "\t"
             << p[i].turnaround << " - " << p[i].burst << " = " << p[i].waiting << "\n";
    }
    cout << "\nAverage Turnaround Time = " << fixed << setprecision(2) << (totalTAT / n);
    cout << "\nAverage Waiting Time = " << fixed << setprecision(2) << (totalWT / n) << "\n";
    return 0;
}
```

Output:

```
C:\csit\fourth_sem_jonash\OS
==================================
||       FCFS Scheduling          ||
|| Compiled by :- Jonash Chataut  ||
==================================

Enter number of processes: 5
Enter process name, arrival time, and burst time:
A 0 10
B 2 15
C 3 22
D 5 16
E 6 5

Gantt Chart:
-------------------------------------------------
|   A   |   B   |   C   |   D   |   E   |
-------------------------------------------------
0       10      25      47      63      68

Process AT      BT      CT      TAT (CT-AT)     WT (TAT-BT)
-----------------------------------------------------------------
A       0       10      10      10 - 0 = 10     10 - 10 = 0
B       2       15      25      25 - 2 = 23     23 - 15 = 8
C       3       22      47      47 - 3 = 44     44 - 22 = 22
D       5       16      63      63 - 5 = 58     58 - 16 = 42
E       6       5       68      68 - 6 = 62     62 - 5 = 57

Average Turnaround Time = 39.40
Average Waiting Time = 25.80
```