VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Principles of Cloud Computing
CSE3035

Lab Assignment 6-
Scheduling Algorithms (FCFS and SJF)

Slot : L11+L12

Name : Kulvir Singh

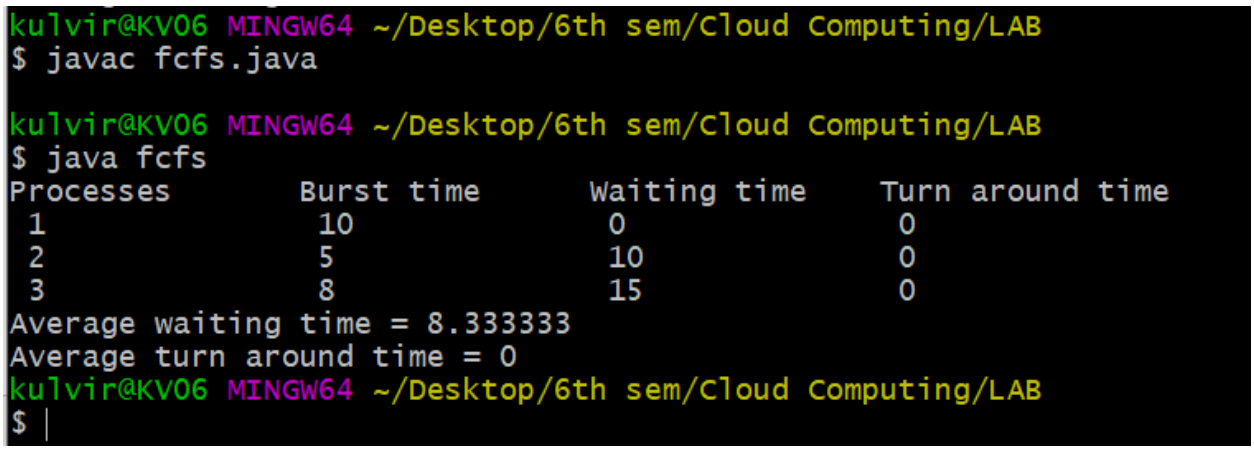
Register Number : 19BCE2074

First Come First Serve :

Code :

```
import java.text.ParseException;
class fcfs {
   static void findWaitingTime(int processes[], int n, int bt[], int wt[]) {
      wt[0] = 0;
      for (int i = 1; i < n; i++) {
         wt[i] = bt[i - 1] + wt[i - 1];
      }
   }
   static void findTurnAroundTime(int processes[], int n, int bt[], int wt[], int tat[]) {
      for (int i = 0; i < n; i++)
         tat[i] = bt[i] + wt[i];
   }
   static void findavgTime(int processes[], int n, int bt[]) {
      int wt[] = new int[n], tat[] = new int[n];
      int total_wt = 0, total_tat = 0;
      findWaitingTime(processes, n, bt, wt);
      System.out.printf("Processes\tBurst time\tWaiting time\tTurn around time\n");
      for (int i = 0; i < n; i++) {
         total_wt = total_wt + wt[i];
         total_tat = total_tat + tat[i];
         System.out.printf(" %d \t\t", (i + 1));
         System.out.printf(" %d \t\t", bt[i]);
         System.out.printf(" %d\t\t", wt[i]);
         System.out.printf(" %d\n", tat[i]);
      }
      float s = (float)total_wt /(float) n;
      int t = total_tat / n;
      System.out.printf("Average waiting time = %f", s);
      System.out.printf("\n");
      System.out.printf("Average turn around time = %d ", t);
   }
   public static void main(String[] args) throws ParseException {
      int processes[] = {1, 2, 3};
      int n = processes.length;
      int burst_time[] = {10, 5, 8};
      findavgTime(processes, n, burst_time);
   }
}
```

Output Screenshot :



Shortest Job First :

Code :

```
import java.util.*;
class sjf{
   static int[][] mat = new int[10][6];
   static void arrangeArrival(int num, int[][] mat)
   {
      for (int i = 0; i < num; i++) {
         for (int j = 0; j < num - i - 1; j++) {
            if (mat[j][1] > mat[j + 1][1]) {
               for (int k = 0; k < 5; k++) {
                  int temp = mat[j][k];
                  mat[j][k] = mat[j + 1][k];
                  mat[j + 1][k] = temp;
               }
            }
         }
      }
   }
   static void completionTime(int num, int[][] mat)
   {
      int temp, val = -1;
      mat[0][3] = mat[0][1] + mat[0][2];
      mat[0][5] = mat[0][3] - mat[0][1];
      mat[0][4] = mat[0][5] - mat[0][2];
      for (int i = 1; i < num; i++) {
         temp = mat[i - 1][3];
         int low = mat[i][2];
         for (int j = i; j < num; j++) {
            if (temp >= mat[j][1] && low >= mat[j][2]) {
               low = mat[j][2];
               val = j;
            }
         }
         mat[val][3] = temp + mat[val][2];
         mat[val][5] = mat[val][3] - mat[val][1];
         mat[val][4] = mat[val][5] - mat[val][2];
         for (int k = 0; k < 6; k++) {
            int tem = mat[val][k];
            mat[val][k] = mat[i][k];
            mat[i][k] = tem;
         }
      }
```

```
    }
    public static void main(String[] args)
    {
        int num;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of Process: ");
        num = sc.nextInt();
        System.out.println("...Enter the process ID...");
        for (int i = 0; i < num; i++) {
            System.out.println("...Process " + (i + 1) + "...");
            System.out.println("Enter Process Id: ");
            mat[i][0] = sc.nextInt();
            System.out.println("Enter Arrival Time: ");
            mat[i][1] = sc.nextInt();
            System.out.println("Enter Burst Time: ");
            mat[i][2] = sc.nextInt();
        }

        System.out.println("Before Arrange...");
        System.out.println(
            "Process ID\tArrival Time\tBurst Time");
        for (int i = 0; i < num; i++) {
            System.out.printf("%d\t\t%d\t\t%d\n", mat[i][0],
                    mat[i][1], mat[i][2]);
        }
        arrangeArrival(num, mat);
        completionTime(num, mat);
        System.out.println("Final Result...");
        System.out.println(
            "Process ID\tArrival Time\tBurst"
            + " Time\tWaiting Time\tTurnaround Time");
        for (int i = 0; i < num; i++) {
            System.out.printf(
                "%d\t\t%d\t\t%d\t\t%d\t\t%d\n", mat[i][0],
                mat[i][1], mat[i][2], mat[i][4], mat[i][5]);
        }
        sc.close();
    }
}
```

Output Screenshot :

```
kulvir@KVO6 MINGW64 ~/Desktop/6th sem/Cloud Computing/LAB
$ javac sjf.java

kulvir@KVO6 MINGW64 ~/Desktop/6th sem/Cloud Computing/LAB
$ java sjf
enter number of processes :
3
...Enter the process ID...
Process 1
enter process id
1
enter arrival time
1
enter burst time
3
Process 2
enter process id
2
enter arrival time
2
enter burst time
4
Process 3
enter process id
3
enter arrival time
5
enter burst time
6
Before Arrange :
process id      arrival time    burst time
1               1               3
2               2               4
3               5               6
Final Result :
Process ID      Arrival Time    Burst Time      Waiting Time    Turnaround Time
1               1               3               -4              -1
3               5               6               0               0
2               2               4               6               10

kulvir@KVO6 MINGW64 ~/Desktop/6th sem/Cloud Computing/LAB
$
```