Name: Maimoona Khilji
Program: BS-Data Science

Operating System – Lab 21

Task 1: Solve FCFS Algorithm.

Code:

FCFS - Notepad

File Edit Format View Help

```
import java.util.*;
public class FCFS {
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("enter no of process: ");
                int n = sc.nextInt();
                int pid[] = new int[n];   // process ids
                int ar[] = new int[n];     // arrival times
                int bt[] = new int[n];     // burst or execution times
                int ct[] = new int[n];     // completion times
                int ta[] = new int[n];     // turn around times
                int wt[] = new int[n];     // waiting times
                int temp;
                float avgwt=0,avgta=0;

                for(int i = 0; i < n; i++)
                {
                        System.out.print("enter process " + (i+1) + " arrival time: ");
                        ar[i] = sc.nextInt();
                        System.out.print("enter process " + (i+1) + " brust time: ");
                        bt[i] = sc.nextInt();
                        pid[i] = i+1;
                }
                        //sorting according to arrival times
                for(int i = 0 ; i <n; i++)
                {
                        for(int  j=0;  j < n-(i+1) ; j++)
                        {
                                if( ar[j] > ar[j+1] )
                                {
                                        temp = ar[j];
                                        ar[j] = ar[j+1];
                                        ar[j+1] = temp;
                                        temp = bt[j];
                                        bt[j] = bt[j+1];
```

Name: Maimoona Khilji

Program: BS-Data Science

FCFS - Notepad

File Edit Format View Help

```
                        temp = bt[j];
                        bt[j] = bt[j+1];
                        bt[j+1] = temp;
                        temp = pid[j];
                        pid[j] = pid[j+1];
                        pid[j+1] = temp;
                }                               }               }
// finding completion times
for(int  i = 0 ; i < n; i++)
{
        if( i == 0)
        {
                ct[i] = ar[i] + bt[i];
        }
        else
        {
                if( ar[i] > ct[i-1])
                {
                        ct[i] = ar[i] + bt[i];
                }
                else
                        ct[i] = ct[i-1] + bt[i];
        }
        ta[i] = ct[i] - ar[i] ;          // turnaround time= completion time- arrival time
        wt[i] = ta[i] - bt[i] ;          // waiting time= turnaround time- burst time
        avgwt += wt[i] ;               // total waiting time
        avgta += ta[i] ;               // total turnaround time
}
System.out.println("\npid  arrival  brust  complete turn waiting");
for(int  i = 0 ; i< n;  i++)
{
        System.out.println(pid[i] + "  \t " + ar[i] + "\t" + bt[i] + "\t" + ct[i] + "\t" + ta[i] + "\t"  + wt[i] ) ;
}
sc.close();
System.out.println("\naverage waiting time: "+ (avgwt/n));     // printing average waiting time.
System.out.println("average turnaround time:"+(avgta/n));    // printing average turnaround time.              }       }
```

Output:

```
maimoona@DESKTOP-3E3NBI6:/mnt/c/Users/Maimoona Khilji$ java FCFS
enter no of process:
5
enter process 1 arrival time: 0
enter process 1 brust time: 10
enter process 2 arrival time: 0
enter process 2 brust time: 6
enter process 3 arrival time: 0
enter process 3 brust time: 2
enter process 4 arrival time: 0
enter process 4 brust time: 4
enter process 5 arrival time: 0
enter process 5 brust time: 8

pid  arrival  brust  complete turn waiting
1        0      10      10      10      0
2        0      6       16      16      10
3        0      2       18      18      16
4        0      4       22      22      18
5        0      8       30      30      22

average waiting time: 13.2
average turnaround time:19.2
```

Name: Maimoona Khilji
Program: BS-Data Science

Task 2: Solve Priority Based Scheduling Algorithm.

Code:

*GFG - Notepad

File Edit Format View Help

```
import java.util.*;
class Process
{
        int pid; // Process ID
        int bt; // CPU Burst time required
        int priority; // Priority of this process
        Process(int pid, int bt, int priority)
        {
                this.pid = pid;
                this.bt = bt;
                this.priority = priority;
        }
        public int prior() {
                return priority;
        }
}
public class GFG
{
        public void findWaitingTime(Process proc[], int n,int wt[])                    // Function to find the waiting time for all processes
        {

                wt[0] = 0;                                                              // waiting time for first process is 0
                for (int i = 1; i < n ; i++ )                                           // calculating waiting time
                        wt[i] = proc[i - 1].bt + wt[i - 1] ;
        }
        public void findTurnAroundTime( Process proc[], int n,int wt[], int tat[])             // Function to calculate turn around time
        {
                for (int i = 0; i < n ; i++)                                            // calculating turnaround time by adding bt[i] + wt[i]
                        tat[i] = proc[i].bt + wt[i];
        }
        public void findavgTime(Process proc[], int n)                                  // Function to calculate average time
        {
                int wt[] = new int[n], tat[] = new int[n], total_wt = 0, total_tat = 0;
                findWaitingTime(proc, n, wt);                                           // Function to find waiting time of all processes
                findTurnAroundTime(proc, n, wt, tat);                                   // Function to find turn around time for all processes
                                                                                        // Display processes along with all details
```

*GFG - Notepad

File Edit Format View Help

```
                                                                                        // Display processes along with all details

                System.out.print("\nProcesses   Priority        Burst time       Waiting time   Turn around time\n");

                for (int i = 0; i < n; i++)     {                                       // Calculate total waiting time and total turn around time
                        total_wt = total_wt + wt[i];
                        total_tat = total_tat + tat[i];
                        System.out.print("    " + proc[i].pid + "\t\t    " + proc[i].priority + "\t\t    "
                                          + proc[i].bt + "\t\t     " + wt[i] + "\t\t     " + tat[i] + "\n");      }

                System.out.print("\nAverage waiting time = " + (float)total_wt / (float)n);
                System.out.print("\nAverage turn around time = "+(float)total_tat / (float)n);
        }
        public void priorityScheduling(Process proc[], int n){          // Sort processes by priority
                Arrays.sort(proc, new Comparator<Process>() {
                        @Override
                        public int compare(Process a, Process b) {
                                return b.prior() - a.prior();
                                }       }               );

                System.out.print("Order in which processes gets executed \n");
                for (int i = 0 ; i < n; i++)
                        System.out.print(proc[i].pid + " ") ;
                        findavgTime(proc, n);
        }
        public static void main(String[] args)          // Driver code
        {
                GFG ob=new GFG();
                int n = 3;
                Process proc[] = new Process[n];
                proc[0] = new Process(1, 10, 2);
                proc[1] = new Process(2, 5, 0);
                proc[2] = new Process(3, 8, 1);
                ob.priorityScheduling(proc, n);
                System.out.print("\n");
        }       }
```

Name: Maimoona Khilji
Program: BS-Data Science

Output: