

Priority scheduling (non-preemptive)-5

```
import java.util.*;
```

```
class PriorityNonPreemptive {
```

```
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.print("Enter number of processes: ");
```

```
        int n = sc.nextInt();
```

```
        int pid[] = new int[n], at[] = new int[n], bt[] = new int[n], pr[] = new int[n];
```

```
        int wt[] = new int[n], tat[] = new int[n], ct[] = new int[n];
```

```
        boolean done[] = new boolean[n];
```

```
        for(int i=0;i<n;i++){
```

```
            System.out.print("Process ID: ");
```

```
            pid[i] = sc.nextInt();
```

```
            System.out.print("Arrival Time: ");
```

```
            at[i] = sc.nextInt();
```

```
            System.out.print("Burst Time: ");
```

```
            bt[i] = sc.nextInt();
```

```
            System.out.print("Priority (lower value = higher priority): ");
```

```
            pr[i] = sc.nextInt();
```

```
            System.out.println();
```

```
        }
```

```
        int time = 0, completed = 0;
```

```
        ArrayList<Integer> order = new ArrayList<>();
```

```
        while(completed < n) {
```

```
            int idx = -1, bestPriority = Integer.MAX_VALUE;
```

```

for(int i=0; i<n; i++){
    if(!done[i] && at[i] <= time && pr[i] < bestPriority){
        bestPriority = pr[i];
        idx = i;
    }
}

if(idx == -1) { time++; continue; } // idle CPU

time += bt[idx];
ct[idx] = time;
tat[idx] = ct[idx] - at[idx];
wt[idx] = tat[idx] - bt[idx];
done[idx] = true;
completed++;
order.add(pid[idx]);
}

double avgWT = 0, avgTAT = 0;
System.out.println("\nPID\tAT\tBT\tPR\tWT\tTAT");
for(int i=0;i<n;i++){
    avgWT += wt[i];
    avgTAT += tat[i];
    System.out.println(pid[i]+"\\t"+at[i]+"\\t"+bt[i]+"\\t"+pr[i]+"\\t"+wt[i]+"\\t"+tat[i]);
}

System.out.println("\nExecution Order:");
for(int p : order) System.out.print("P"+p+" ");

System.out.println("\n\nAverage Waiting Time: " + avgWT/n);

```

```
        System.out.println("Average Turnaround Time: " + avgTAT/n);  
    }  
}
```