PRIORITY (PREMPTION):

CODE :

import java.util.Scanner;

public class PriorityPreemptive {

static class Process {

int id, burst, priority, arrival, waitingTime, turnaroundTime;

int remainingBurst;

public Process(int id, int burst, int priority, int arrival) {

this.id = id;

this.burst = burst;

this.priority = priority;

this.arrival = arrival;

this.remainingBurst = burst;

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int n = sc.nextInt();

Process[] processes = new Process[n];

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

System.out.print("Enter arrival time, burst time, and priority for process " + (i + 1) + ": ");

int at = sc.nextInt();

int bt = sc.nextInt();

int priority = sc.nextInt();

processes[i] = new Process(i + 1, bt, priority, at);

}

int completed = 0, time = 0;

String ganttChart = "";

while (completed < n) {

int idx = -1;

int highestPriority = Integer.MAX\_VALUE;

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

if (processes[i].arrival <= time && processes[i].remainingBurst > 0 && processes[i].priority < highestPriority) {

highestPriority = processes[i].priority;

idx = i;

}

}

if (idx != -1) {

processes[idx].remainingBurst--;

ganttChart += "P" + processes[idx].id + " ";

if (processes[idx].remainingBurst == 0) {

completed++;

processes[idx].turnaroundTime = time + 1 - processes[idx].arrival;

processes[idx].waitingTime = processes[idx].turnaroundTime - processes[idx].burst;

}

} else {

ganttChart += "idle ";

}

time++;

}

// Print Gantt Chart

System.out.println("Gantt Chart: " + ganttChart);

// Calculate and print average waiting time and turnaround time

int totalWT = 0, totalTAT = 0;

System.out.println("Process\tArrival\tBurst\tPriority\tWaiting\tTurnaround");

for (Process p : processes) {

totalWT += p.waitingTime;

totalTAT += p.turnaroundTime;

System.out.println("P" + p.id + "\t" + p.arrival + "\t" + p.burst + "\t" + p.priority + "\t\t" + p.waitingTime + "\t" + p.turnaroundTime);

}

System.out.println("Average Waiting Time: " + (totalWT / (float) n));

System.out.println("Average Turnaround Time: " + (totalTAT / (float) n));

sc.close();

}

}

OUTPUT :

Enter number of processes: 5

Enter arrival time, burst time, and priority for process 1: 10 2 1

Enter arrival time, burst time, and priority for process 2: 10 2 2

Enter arrival time, burst time, and priority for process 3: 11 1 3

Enter arrival time, burst time, and priority for process 4: 13 3 4

Enter arrival time, burst time, and priority for process 5: 14 1 5

Gantt Chart: idle idle idle idle idle idle idle idle idle idle P1 P1 P2 P2 P3 P4 P4 P4 P5

Process Arrival Burst Priority Waiting Turnaround

P1 10 2 1 0 2

P2 10 2 2 2 4

P3 11 1 3 3 4

P4 13 3 4 2 5

P5 14 1 5 4 5

Average Waiting Time: 2.2

Average Turnaround Time: 4.0