

Question 6

Write a program to implement the Non-preemptive priority scheduling algorithm and find the average turnaround time, waiting time, completion time and response time for overall process. Also Print Gantt chart for it.

Garv nanwani

19BCS049

CODE :

```
#include<stdio.h>
#include<string.h>
void sort(int arr[][7],char str[][10], int at, int bt, int pr, char p[], int n, int m);
void print(int n, char str[][10], int arr[][7] );
void ganttChart(int time[],char gantt[][10], int m, int l);

int main(){
    char process[10], gantt[100][10];
    int time[100];
    int at, bt, n, pr;
    printf("Enter no of process :");
    scanf("%d",&n);
    int arr[n+1][7];
    int temp[n];
    char str[n][10];

    printf("Enter 'process priority arrival_time burst_time' :\n");
    scanf("%s",str[0]);
    scanf("%d",&arr[0][0]);
    scanf("%d",&arr[0][1]);
    scanf("%d",&arr[0][2]);
    for (int i=1; i<n; i++){
        scanf("%s",process);
        scanf("%d",&pr);
        scanf("%d",&at);
        scanf("%d",&bt);
        int j=0;
        while (j<i && arr[j][1]<=at){
            j++;
        }
        sort(arr,str,at,bt,pr,process,i,j);
    }

    for (int i=0; i<n; i++){
        arr[i][6]=-1;
        temp[i]=arr[i][2];
    }

    time[0]=arr[0][1];
    int l=1, m=0, cnt=0, t=0;
```

```

arr[n][0]=10000;

while (cnt<n){
    int min=n;
    bool flag=false;
    for (int i=0; i<n; i++){
        if (arr[i][1]<=t && temp[i]>0 && arr[i][0]<arr[min][0]){
            min=i;
            flag=true;
        }
    }

    if (flag){
        arr[min][3]=t+arr[min][2];
        arr[min][4]=arr[min][3]-arr[min][1];
        arr[min][5]=arr[min][4]-arr[min][2];
        arr[min][6]=t-arr[min][1];
        temp[min]=0;
        t+=arr[min][2];
        time[l]=t;
        l++;
        strcpy(gantt[m],str[min]);
        m++;
        cnt++;
    }
    else{
        int num=0;
        for (int i=0; i<n; i++){
            if (temp[i]>0){
                num=i;
                break;
            }
        }
        t=arr[num][1];
        time[l]=t;
        l++;
        strcpy(gantt[m],"lag");
        m++;
    }
}

print(n,str,arr);
ganttChart(time,gantt,m,l);
return 0;
}

void sort(int arr[][7],char str[][10], int at, int bt, int pr, char p[], int n, int m)
{
    for (int i=n-1; i>=m; i--){
        arr[i+1][0]=arr[i][0];
        arr[i+1][1]=arr[i][1];
        arr[i+1][2]=arr[i][2];
    }
}

```

```

        strcpy(str[i+1],str[i]);
    }
    arr[m][0]=pr;
    arr[m][1]=at;
    arr[m][2]=bt;
    strcpy(str[m],p);
}

void print(int n, char str[][10], int arr[][7] ){
    float avg;
    float sum;
    char title[8][20]={"Process","Priority","Arrival Time","Burst Time","Completion
Time","T.A.T",
                        "Waiting Time","Response Time"};

    printf("\n\n");
    for (int i=0; i<8; i++){
        printf("%-20s",title[i]);
    }
    printf("\n");
    for (int i=0; i<n; i++){
        printf("%-20s",str[i]);
        for (int j=0; j<7; j++){
            printf("%-20d",arr[i][j]);
        }
        printf("\n\n");
    }
    printf("%-80s", "Average");
    for (int j=3; j<7; j++){
        sum=0;
        for (int i=0; i<n; i++){
            sum+=arr[i][j];
        }
        avg=sum/n;
        printf("%-20.2f", avg);
    }
    printf("\n\n");
}

void ganttChart(int time[],char gantt[][10], int m, int l){
    printf("Gantt Chart :\n\n");
    printf("|");
    for (int i=0; i<m; i++){
        printf("%-5s|",gantt[i]);
    }
    printf("\n\n");
    for (int i=0; i<l; i++){
        printf("%-6d",time[i]);
    }
}

```

Output :

Enter no of process :5

Enter 'process priority arrival_time burst_time' :

p1 2 0 5

p2 5 2 4

p3 4 4 6

p4 6 4 5

p5 4 6 3

Process	Priority	Arrival Time	Burst Time	
Completion Time	T.A.T	Waiting Time	Response Time	
p1	2	0	5	5
5	0	0		
p2	5	2	4	18
16	12	12		
p3	4	4	6	11
7	1	1		
p4	6	4	5	23
19	14	14		
p5	4	6	3	14
8	5	5		
Average				14.20
11.00	6.40	6.40		

Gantt Chart :

p1	p3	p5	p2	p4	
0	5	11	14	18	23

```
Enter no of process :5
Enter 'process priority arrival_time burst_time' :
p1 2 0 5
p2 5 2 4
p3 4 4 6
p4 6 4 5
p5 4 6 3

Process      Priority  Arrival Time  Burst Time  Completion Time  T.A.T    Waiting Time  Response Time
p1           2        0             5           5              5        0            0
p2           5        2             4           18             16       12           12
p3           4        4             6           11             7        1            1
p4           6        4             5           23             19       14           14
p5           4        6             3           14             8        5            5
Average                        14.20         11.00       6.40        6.40

Gantt Chart :
|p1 |p3 |p5 |p2 |p4 |
0   5   11  14  18  23
```