13

28/6/23

Lab-3

Write a program to schedule non preemptive priority queue.

```
#include <stdio.h>
#include <stdlib.h>
int at[10], t, pt[10], tat[10], wt[10], n, time = 0, ready[10], py[10],
float atat = 0, awt = 0;
void main () {
  printf("Enter number of process \n");
  scanf("%d", &n);
  printf("Enter arrival time : \n");
  for(i=0; i<n; i++)
  scanf("%d", &at[i]);

  printf("Enter process time: \n");
  for(i=0; i<n; i++)
   scanf("%d", &pt[i]);

  printf("Enter priority: \n");
  for(i=0; i<n; i++)
   scanf("%d", &py[i]);

  for(i=0; i<n; i++)
   ready[i] = 0;

  for(i=0; i<n; i++)
   p[i] = pt[i];

  for (i=0; i<n; i++)
     time += pt[i];
     t = n;
```

op = 0, max, 3,
p[10];

```
while(t--){
for(i=0;i<n;i++)
if(op>=at[i])
ready[i]=1;
for(i=0;i<n;i++)
if(bt[i]==0)
pry[i]=0;
maxpri=pry[0];
for(i=0;i<n;i++)
if(ready[i]==1
if(pry[i]>maxpri)
maxpri=pry[i];
for(i=0;i<n;i++)
if(maxpri==pry[i])
x=i;
printf("%d %d",op,(x+1));
op=op+bt[x]
tat[x]=op;
ready[x]=0;
pry[x]=0;
}
printf("%d",op);
```

```
for(i=0,i<n;i++){
tat[i]=tat[i]-at[i]
}
for(i=0;i<n;i++){
atat+=tat[i]
wt[i]=tat[i]-bt[i];
}
for(i=0;i<n;i++){
aut+=wt[i];
awt=awt/n;
atat=atat/n;
printf("\n");
for(i=0;i<n;i++)
printf("%d %d %d %d
printf("ATAT=%f Tn
```

output:

Enter the no of Process: 4
Enter the arrival time: 0 1 2 3
Enter the Process time: 4 3 3 2
Enter the Priority: 3 4

P_1	P_3	P_4	P_2
0	4	7	12

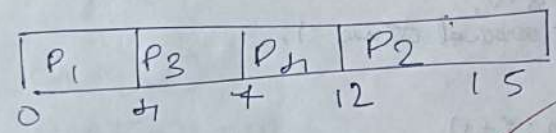
P_1	4	0
P_2	12	11
P_3	5	2
P_4	9	4

ATAT = 8.00s
AWT = 4.2500s

```
for (i=0; i<n; i++) {
    tat[i] = tat[i] - at[i]
}
for(i=0; i<n; i++){
    atat += tat[i]
    wt[i] = tat[i] - bt[i];
}
for(i=0; i<n; i++){
    awt += wt[i];
    atat = atat/n;
    awt = awt/n;
printf("\n");
for (i=0; i<n; i++)
    printf("%d %d %d %d\n", i+1, tat[i], wt[i]);
printf("ATAT = %f\n awt = %f", atat, awt);
```

Output:

Enter the no of Process: 4
Enter the arrival time: 0 1 2 3
Enter the Process time: 4 3 3 5
Enter the Priority: 3 4 6 5



P_1	4	0
P_2	14	11
P_3	5	2
P_4	9	4

ATAT = 8.00s
AWT = 4.250s

Write a C program to schedule the process in round robin

```
#include<stdio.h>
int at[10], bt[10], p[10], time=0, op=0, i, j, n;
int ready[10], q[100], r=-1, f=0, tat[10], wt[10], z;
int y=9999, dq, tq;
float att, awt;
int rr(int n){
  if (bt[z] > tq){
    bt[z] -= tq;
    op += tq;
  }
  else{ op = bt[z];
        bt[z] = 0;
        tat[z] = op;
        ready[z] = 0;
  }
  return z;
}

void main(){
 printf("Enter number of process : \n");
 scanf("%d", &n);
 printf("Enter arrival time \n");
 for(i=0; i<n; i++)
    scanf("%d", &at[i]);
 printf("\n Enter the process time:");
 for(i=0; i<n; i++)
    scanf("%d", &bt[i]);
 printf("Enter time quantum \n");
 scanf("%d", &tq);
 for(i=0; i<n; i++)
    ready[i]=0;
 for(i=0; i<n; i++)
    q[i] = 9999;
 for(i=0; i<n; i++)
   time += bt[i];
 for(i=0; i<n; i++)
    ready[i] = 1;
 for(i=0; i<n; i++)
   if(ready[i] == 1)
   { q[++r]=i;
   }
 while(op != time)
 { printf("\n%d", op);
   if(z==y)
    q[++f]
    y = z;
   ch=q[f]
   if(bt[n] != 0)
   { z = rr(q[f]);
    printf("P %d",
    for (i=0; i
```

round robin

```
=0, i, j, n;
[0], z;
```

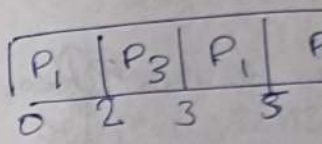
```
printf("Enter TQ\n");
scanf("%d", &tq);
for(i=0; i<n; i++)
  ready[i]=0;
for(i=0; i<n; i++)
  q[i] = 9999;
for(i=0; i<n; i++)
  time += pt[i];
for(i=0; i<n; i++)
  ready[i] = 1;
for(i=0; i<n; i++)
  if(ready[i] == 1)
  { q[++r]=i;
  }
while(op1 <= time)
{ printf("\n%d", op);
  if(z==y)
    q[++f]
    y = z;
  ch = q[f]
  if(pt[n] != 0)
  { z = rr(q[f]);
    printf("P %d", (z+1));
  for(i=0; i<n; i++){
```

```
if (cp >= at[i] && bt[i] != 0) {
    flg = 0;
    j = f;
    while (j <= r)
    { if (i == q[j])
        flg = 1;
      j++;
    }
    if (flg == 0)
      q[++r] = i;
    }
  if (bt[i] != 0)
      q[++r] = i;
    }
  f++;
}

for (i=0; i<n; i++) {
    tat[i] = tat[i] - at[i];
    wt[i] = bt[i] - p[i];
    atat += tat[i];
    awt += wt[i];
}

atat = atat/n;
awt = awt/n;
```

```
printf("\n PID \t AT \t BT \t CT \t TAT \t WT");
printf("\n ATAT = %f \n AWT = %f", atat, awt);
}
```

Output

Enter the no of pro
Enter arrival ti
Enter process time
Enter TQ 2



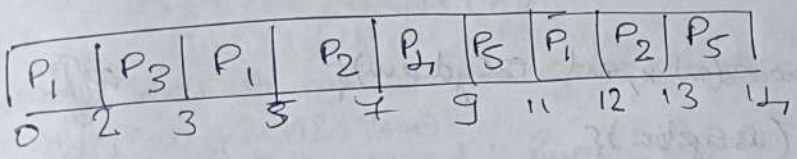
P_1	12	1
P_2	12	9
P_3	1	0
P_4	0	4
P_5	10	7

Average TAT = 8
Avg WT = 5.

```
printf("\n%d\t%d\t%d\t%d\n", (i+1), tat[i], wt[i]);
printf("ATAT = %f\n AWT = %f ", atat, awt);
}
```

Output

Enter the no of process: 5
Enter arrival time 0 1 2 3 4
Enter process time 5 3 1 2 3
Enter TQ 2



P_1	12	7
P_2	12	9
P_3	1	0
P_4	0	2
P_5	10	7

3/7/23

Average TAT = 8.025.

Avg WT = 5.2