classmate
Date 28th June
Page 16

Write a c-program to simulate the CPU scheduling algorithms round-robin and priority.

Priority (Non-premtive)

```
#include <stdio.h>
#include <conio.h>
int main ()
{
int n, i, bt[20], priority[20], at[20], j, temp, wt[20],
tat[20], sum=0, avgwt=0, avgtat=0;
Printf ("Enter no. of processors:");
Scanf ("%d", &n);
for (i=0; i<n; i++)
printf ("Enter arrival time for p[%d]:", i+1);
Scanf ("%d", &at[i]);
Printf ("Enter burst time for p[%d]:", i+1);
Scanf ("%d", &bt[i]);
Printf ("Enter priority of p[%d]:", i+1);
Scanf ("%d", &priority[i]);
}
for (i=0; i<n-1; i++)
for (j=0; j<n-1-i; j++)
if (priority[j] > priority[j+1])
temp = priority[j];
priority[j] = priority[j+1];
priority[j+1] = temp;
temp = bt[j];
bt[j] = bt[j+1];
bt[j+1] = temp;
temp = at[j];
at[j] = at[j+1];
at[j+1] = temp;
}
}
}
```

```
for(i=0; i<n; i++){
wt[i] = sum - at[i];
tat[i] = wt[i] + bt[i];
printf("%d\n", wt[i]);
avgwt += wt[i];
avgtat += tat[i];
sum += bt[i];
}
float avgwt_f = (float)avgwt/n;
float avgtat_f = (float)avgtat/n;
Printf("\n Total average waiting time: %f ", avgwt_f);
Printf("\n Total average turnaround time: %f", avgtat_f);
getch();
return 0;
}
```

lower no = higher Priority.

P1	P1	P1	P1	P4	P3	P2
0	1	2	3	4	9	12	15

Output

```
Enter no. of process: 4
Enter arrival time for p[1]: 0
   "   burst   "   "   "  : 4
   "   Priority  "   "   : 1
Enter arrival time for p[2]: 1
   "   burst   "   "   "  : 3
   "   Priority  "   "   : 4
Enter arrival time for p[3]: 2
   "   burst time  "   "  = 3
Enter  Priority   "   "  = 3
Enter  arrival    "   "  = 3
   "   burst     "   "  = 5
   "   Priority  "   "  = 2.
```

P1(3) P1(2) P1(1) P2(3) P2(3) P2(3)
P2(3) P2(3) P2(3) P3(3) P3(3)
P3(3) P3(3) P4(5)
P4(5)

Process	AT	BT	Priority	TAT	WT
P[1]	0	4	1	4	0
P[4]	3	5	2	6	1
P[3]	2	3	3	10	7
P[2]	1	3	4	14	11

Total average waiting time: 4.75.
Total Turnaround time = 8.5

```
Enter number of processors: 4
Enter arrival time for p[1]: 0
Enter burst time for p[1]: 4
Enter priority of p[1]: 3
Enter arrival time for p[2]: 1
Enter burst time for p[2]: 3
Enter priority of p[2]: 4
Enter arrival time for p[3]: 2
Enter burst time for p[3]: 3
Enter priority of p[3]: 66
Enter arrival time for p[4]: 3
Enter burst time for p[4]: 5
Enter priority of p[4]: 5
0
3
4
10

Total average waiting time: 4.250000
Total average turnaround time: 8.000000
```

Round - Robin.

```
#include <stdio.h>
int tq, at[10], pt[10], p[10], tim=0, op=0, i, j, n,
ready[10], q[100];
int r=-1, f=0, tat[10], wt[10], x, ab, y=9999, ch;
int float atat, awt;
int roundrobin(int x)
{
 if(pt[n]>tq)
 {
 pt[n] -= tq;
  op+=tq;
 }
else
 {
 op+=pt[n];
  pt[n]=0;
 tat[n]=op;
 ready[n]=0;
 }
return x;
}
void main()
{
Printf("Enter num. of processes\n"); scanf("%d", &n);
for(i=0; i<n; i++)
scanf("%d", &at[i]);
Printf("Enter arrival time\n");
Printf("Enter process time : \n");
for(i=0; i<n; i++)
Scanf("%d", &pt[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++)
  p[i]=pt[i];
for(i=0;i<n;i++)
  time+=pt[i];
for(i=0;i<n;i++)
if(ready[i]==1)
{
 q[++r]=i;
 }
while(op!=time)
{
printf("%d",op);
if(x==y)
q[++f];
 y=x;
ch=q[f];
if(pt[ch]!=0)
{
x=rr(q[f]);
printf("p%d",(x+1));
for(i=0;i<n;i++)
{
if(op>=at[i]&&pt[i]!=0)
{
  ab=0;
  j=f;
while(j<=r)
{
if(i==q[j])
fg=1;
j++;
}
```

```
}
if (ab == 0)
{
 q[++r] = i;
}
}
}
if (pt[z] != 0)
 q[++r] = z;
}
f++;
}
Printf("%d", op);
for (i=0; i<n; i++)
{
tat[i] = tat[i] - at[i];
wt[i] = tat[i] - p[i];
awt += wt[i];
atat += tat[i];
}
atat = atat/n;
awt = awt/n;
Printf("\n");
for (i=0; i<n; i++)
Printf("P %d %d %d \n", i+1, tat[i], wt[i]);
Printf("ATAT = %f \n AWT = %f", atat, awt);
}
```

Output

```
Enter no. of process : 4
Enter arrival time for p[1]: 0
 "   process "  "  "  : 5
Enter arrival time for p[2]: 1
 "  Process "  "  p[2]: 4
```

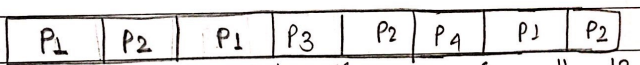
Enter arrival time for P[3]: 2
" Process " " P[3]: 2
Enter arrival time for P[4]: 4
" Process " " P[4]: 1
Enter Time quantum: 2

~~Process AT PT TAT WT~~
~~P[1] P[3] 0 5 4~~
~~P[2] P[4] 1 4 3~~
~~P[3] P[2] 2 2~~
~~P[4] P[1] 4 1~~

Process	AT	PT	TAT	WT
P[3]	2	2	4	2
P[4]	4	1	3	2
P[2]	1	4	10	6
P[1]	0	5	12	7

ATAT = ~~4.25~~ 7.25
AWT = 4.25

10/10



P1[5] P2[4] P1[1] P3[2] P2[3] P4[1] P1[2]
P1[4] P3[2] P2[3] P4[1] P1[2] P2[1]
P2[3] P4[1] P1[2] P2[1]
P1[2]

N
28/6/23

C:\Users\STUDENT\Desktop\rr1.exe

```
Enter number or processes
3
Enter araival times:
1
3
0
Enter process times:
5
2
6
Enter TQ
2
0 P3 2 P1 4 P3 6 P2 8 P1 10 P3 12 P1 13
P1 12 7
P2 5 3
P3 12 6
ATAT=9.666667
AWT=5.333333
Process returned 28 (0x1C)   execution time : 69.047 s
Press any key to continue.
```