LAB 2

Write a C program to simulate the following CPU scheduling algorithm to find turnaround time and waiting time.

Priority (pre-emptive or Non-pre-emptive)

Round Robin (Experiment with different quantum sizes for RR algorithm)

28/6/23

Lab 3:

Write a C program to simulate the following CPU scheduling algorithm to find turnaround time & waiting time.

→ Priority (Pre-emptive & Non-preemptive)

→ Round Robin (Experiment with different quantum sizes for RR algorithm)

1)

```
#include <stdio.h>
#include <conio.h>

int main() {
  int n, i, bu[10], wa[10], tat[10], t, ct[10], at[10],
  max;
  float awt=0, att=0, temp=0;
  printf("Enter the number of processes: ");
  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", &bu[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;
```

following
around

turn size

at[10],

```
        temp = bt[j];
        bt[j] = bt[j+1];
      bt[j+1] = temp;
      temp = at[j];
    at[j] = 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 avgwtf = (float) avgwt / n;
float avgtatf = (float) avgtat / n;

printf("\nTotal average waiting time: %f", avgwtf);
printf("\nTotal average turnaround time: %f", avgtatf);
getch();

return 0;
}
```

```
;)
#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 number of processes: ");
  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 avgwtf = (float) avgwt / n;
    float avgtatf = (float) avgtat / n;

    printf("\nTotal average waiting time: %f", avgwtf);
    printf("\nTotal average turnaround time: %f", avgtat
    getch();

    return 0;
}
```

Output:

1)

```
Enter the number of processes: 3
P[1]: arrival time: 1
      burst time: 1
P[2]: arrival time: 5
      burst time: 6
P[3]: arrival time: 4
      burst time: 2

time slice: 2
```

Avg turnaround time is : 5.00
Avg waiting time is : -2.333

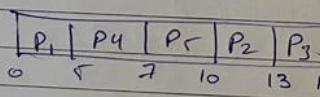
Process	Arrival time	burst time	waiting time	Turnaround ti.
1	4	1	-4	1
2	5	6	-2	9
3	4	2	-1	5

2)

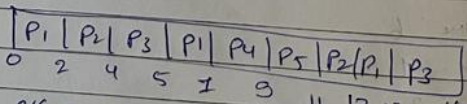
No. of process : 3

Procan	Arrivaltime	burst time	Priority	waiting time
1	1	4	3	-4
2	6	4	4	4
3	4	5	1	2

avg waiting Time : 1.000
avg turnaroud time : 5.333



$P_2(3)$ $P_2(3)$ $P_2(3)$ $P_3(1)$
$P_3(1)$ $P_3(1)$ $P_3(1)$
$P_4(2)$ $P_5(3)$
$P_5(3)$



$P1(3)$ $P3(1)$ $P3(3)$ $P5(3)$ $P2(1)$ $P5(1)$
$P3(1)$ $P1(1)$ $P4(2)$ $P2(1)$ $P1(1)$
$P1(3)$ $P4(2)$ $P5(3)$ $P1(1)$ $P2(1)$
$P5(3)$ $P2(1)$
$P2(1)$

PRIORITY OUTPUT:

```
PS D:\VS Code\OS> cd "d:\VS Code\OS\" ; if ($?) { gcc npp.c -o npp } ; if ($?) { .\npp }
Enter number of processes
4
Enter araival times:
0 1 2 3
Enter process times:
4 3 3 5
Enter priority:
3 4 6 5
0 p1 4 p3 7 p4 12 p2 15
P1  4  0
P2  14  11
P3  5  2
P4  9  4
ATAT=8.000000
AWT=4.250000
PS D:\VS Code\OS>
```

ROUND ROBIN OUTPUT:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL                                  Code

PS D:\VS Code> cd "d:\VS Code\OS\" ; if ($?) { gcc RR1.c -o RR1 } ; if ($?) { .\RR1 }
Enter number or processes
5
Enter araival times:
0 1 2 3 4
Enter process times:
5 3 1 2 3
Enter TQ
2
0 P1 2 P3 3 P1 5 P2 7 P4 9 P5 11 P1 12 P2 13 P5 14
P1  12  7
P2  12  9
P3  1  0
P4  6  4
P5  10  7
ATAT=8.200000
AWT=5.400000
PS D:\VS Code\OS>
```