

6/6/24

Write a C program to simulate multi-level queue scheduling algorithm considering the following scenario. All the processes in the system are divided into two categories - system processes and user processes. System processes are to be given higher priority than user processes. Use FCFS scheduling for the processes in each queue.

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

```
void main()
```

```
{
```

```
    int p[20], bt[20], su[20], wt[20], tat[20],  
        at[20], ct[20], i, k, n, temp;
```

```
    float wtavg, tatavg;
```

```
    printf("Enter the number of processes : ");
```

```
    scanf("%d", &n);
```

```
    for(i=0; i<n; i++)
```

```
    {
```

```
        p[i] = i;
```

```
        printf("Enter the Arrival Time of Process %d", i);
```

```
        scanf("%d", &at[i]);
```

```
        printf("Enter the Burst Time of Process %d", i);
```

```
        scanf("%d", &bt[i]);
```

```
        printf("System/User process (0/1)? ");
```

```
        scanf("%d", &su[i]);
```

```
    }
```

```
    for(i=0; i<n; i++)
```

```
    {
```

```
        for(k=i+1; k<n; k++)
```

```
        {
```

```

if (a[i] > a[k] || (a[i] == a[k] && su[i] > su[k]))
{

```

```

    temp = p[i];

```

```

    p[i] = p[k];

```

```

    p[k] = temp;

```

```

    temp = b[i];

```

```

    b[i] = b[k];

```

```

    b[k] = temp;

```

```

    temp = su[i];

```

```

    su[i] = su[k];

```

```

    su[k] = temp;

```

```

    temp = a[i];

```

```

    a[i] = a[k];

```

```

    a[k] = temp;

```

```

}
}

```

```

wtavg = wt[0] = 0;

```

```

tatavg = tat[0] = b[0];

```

```

ct[0] = a[0] + b[0];

```

```

for (i = 1; i < n; i++)
{

```

```

    if (ct[i-1] < a[i])
    {

```

```

        ct[i] = a[i] + b[i];

```

```

    else
    {

```

```

        ct[i] = ct[i-1] + b[i];

```

```

    wt[i] = ct[i] - a[i] - b[i];

```

```

    tat[i] = ct[i] - a[i];

```



```

        wtavg += wt[i];
        tat[i] = at[i] - at[i];
        wt
        tatavg += tat[i];
    }

    printf("\n PROCESS \t \t ARRIVAL TIME \t \t SYSTEM / USER PROCESS \t \t BURST TIME \t \t WAITING TIME \t \t TURN AROUND TIME");
    for(i=0; i<n; i++)
    {
        printf("\n %d \t \t %d \t \t %d \t \t %d \t \t %d \t \t %d \t \t %d \t \t %d",
            i, at[i], bt[i], wt[i], tat[i], su[i],
            bt[i], wt[i], tat[i]);
    }

    printf("\n Average Waiting Time is %f", wtavg/n);
    printf("\n Average Turn Around Time is %f", tatavg/n);
}

```

Output :-

```

Enter the number processes : 4
Enter the Arrival Time of Process 0 : 0
Enter the Burst Time of Process 0 : 2
system / User Process (0/1) ? : 0
Enter the Arrival Time of Process 1 : 0
Enter the Burst Time of Process 1 : 1
system / User Process (0/1) ? : 1
Enter the Arrival Time of Process 2 : 0
Enter the Burst Time of Process 2 : 5
system / User Process (0/1) ? : 0
Enter the Arrival Time of Process 3 : 0
Enter the Burst Time of Process 3 : 3
system / User Process (0/1) ? : 1

```

PROCESS	ARRIVAL TIME	SYSTEM/USER PROCESS	BURST TIME	waiting Time	Turnaround Time
0	0	0	2	0	2
2	0	0	5	2	7
1	0	1	1	7	8
3	0	1	3	8	11

Average waiting Time is : 4.250000

Average Turnaround Time is : 7.000000

✓
Surya
6/6/21