16/5/2024

SURYA Gold
Date Page

Write a C Program to simulate the following non pre-emptive CPU scheduling algorithm to find turnaround time and waiting

1) FCFS
2) SJF (Non-preemptive)

1) //Program for FCFS

```
#include<stdio.h>
void main()
{
    int p[10], at[10], bt[10], ct[10], tat[10], wt[10],
    i, j, temp=0, n;
    float awt=0, atat=0;
    printf("Enter no of processes:");
    scanf("%d", &n);
    printf("enter %d process:", n);
    for(i=0; i<n; i++)
    {
        scanf("%d", &p[i]);
    }
    printf("enter %d arrival time:", n);
    for(i=0; i<n; i++)
    {
        scanf("%d", &at[i]);
    }
    printf("enter %d burst time:", n);
    for(i=0; i<n; i++)
    {
        scanf("%d", &bt[i]);
    }
    ct[0] = at[0] + bt[0];
```

```
for (i=1; i<n; i++)
{
    temp=0;
    if (ct[i-1]<at[i])
    {
        temp = at[i] - ct[i-1];
    }
    ct[i] = ct[i-1] + bt[i] + temp;
}
printf("\n p\t A.T \t B.T\t C.T\t TAT\t WT");
for(i=0; i<n; i++)
{
    tat[i] = ct[i] - at[i];
    wt[i] = tat[i] - bt[i];
    atat = tat[i];
    awt = wt[i];
}
atat = atat/n;
awt = awt/n;
for(i=0; i<n; i++)
{
    printf("\n P %d \t %d \t %d \t %d \t %d \t %d",
            p[i], at[i], bt[i], ct[i], tat[i], wt[i]);
}
printf("\n average turnaround time is %f", atat);
printf("\n average waiting time is %f", awt);
```

Output :-

```
enter no of processes : 4
enter 4 process : 1 2 3 4
enter 4 arrival time : 0 1 5 6
enter 4 burst time : 2 2 3 4
```

P	A.T	B.T	C.T	TAT	WT
P1	0	2	2	2	0
P2	1	2	4	3	1
P3	5	3	8	3	0
P4	6	4	12	6	2

average turnaround time is 3.500000
average waiting time is 0.750000

2) // Program for SJF (Non-Preemptive)

```
#include <stdio.h>
void main()
{
    int i,n,p[10]={1,2,3,4,5,6,7,8,9,10},min,k=1,btime=0;
    int bt[10],temp,j,at[10],wt[10],tt[10],ta=0,sum=0;
    float wavg=0,tavag=0,tsum=0,wsum=0;
    printf("\nEnter the No. of processes: ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter the burst time of %d process:\n",i+1);
        scanf("%d",&bt[i]);
        printf("Enter the arrival time of %d process:\n",i+1);
        scanf("%d",&at[i]);
    }
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            if(at[i]<at[j])
            {
                temp=p[j];
                p[j]=p[i];
```

```
            p[i] = temp;
            temp = at[j];
            at[j] = at[i];
            at[i] = temp;
            temp = bt[j];
            bt[j] = bt[i];
            bt[i] = temp;
        }
    }
}
for(j=0; j<n; j++)
{
    btime = btime + bt[j];
    min = bt[k];
    for(i=k; i<n; i++)
    {
        if(btime >= at[i] && bt[i] < min)
        {
            temp = p[k];
            p[k] = p[i];
            p[i] = temp;
            temp = at[k];
            at[k] = at[i];
            at[i] = temp;
            temp = bt[k];
            bt[k] = bt[i];
            bt[i] = temp;
        }
    }
    k++;
}
wt[0] = 0;
```

```
for(i=1;i<n;i++)
{
    sum = sum + bt[i-1];
    wt[i] = sum - at[i];
    wsum = wsum + wt[i];
}
wavg = (wsum/n);
for(i=0;i<n;i++)
{
    ta = ta + bt[i];
    tt[i] = ta - at[i];
    tsum = tsum + tt[i];
}
tavg = (tsum/n);
printf("\nProcess \t Burst \t Arrival \t waiting \t
        Turn-around");
for(i=0;i<n;i++)
{
    printf("\n p-%d \t %d \t %d \t %d \t %d",p[i],bt[i],at[i],wt[i],tt[i]);
}
printf("Average waiting time: %f \n", wavg);
printf("Average turn around time: %f", tavg);
}
```

Output :-

Enter the NO. of processes: 4
Enter the burst time of 1 process: 3
Enter the arrival time of 1 process: 0
Enter the burst time of 2 process: 6
Enter the arrival time of 2 process: 1
Enter the burst time of 3 process: 4
Enter the arrival time of 3 process: 4

Enter the burst time of 4 process : 2
Enter the arrival time of 4 process : 6

Process	Burst	Arrival	waiting	Turn-Around
p1	3	0	0	3
p2	6	1	2	8
p4	2	6	3	5
p3	4	4	7	11

Average waiting time : 3.000000
Average Turn around time : 6.750000

16/5/2024