Solution : TP n°3: Task scheduling

Exercise n°1

1) In this exercise you will be requested to implement the FCFS scheduling algorithm and calculate the waiting time and turnaround time for each process. It is assumed that all processes have arrived at the same time and the execution order of the processes is as follow: P_1, P_2, P_3, …..

   For the FCFS scheduling algorithm, read the number of processes in the system and their CPU execution times. Scheduling is carried out on the basis of the arrival time of the processes, independently of their other parameters. Each process will be executed according to its arrival time.

   FCFS is executed using the following algorithm:

```
1. Begin
2. Declare the array size wt[10], bt[10] and tt[10] where:
   - wt: is an array containing the waiting time of each process,
   - bt: is an array containing the execution time of each process,
   - tt: is an array containing the turnaround time of each process,
3. Read the number of processes to be inserted;
4. Read the burst times (execution times) of processes;
5. Calculate the waiting time of each process:
   - wt[i+1]=bt[i]+wt[i]
6. Calculate the turnaround time of each process
   - tt[i+1]=tt[i]+bt[i+1]
7. Calculate the average waiting time and average turnaround time;
8. Display the values;
9. End
```

```
#include<stdio.h>
void main()
{
 int i,j,bt[10],n,wt[10],tt[10],w1=0,t1=0;
 float aw,at;

 printf("enter no. of processes:\n");
 scanf("%d",&n);
 printf("enter the burst time of processes:");
 for(i=0;i<n;i++)
 scanf("%d",&bt[i]);
 for(i=0;i<n;i++)
 {
 wt[0]=0;
 tt[0]=bt[0];
 wt[i+1]=bt[i]+wt[i];
 tt[i+1]=tt[i]+bt[i+1];
 w1=w1+wt[i];
 t1=t1+tt[i];
 }
 aw=w1/n;
 at=t1/n;
 printf("\nbt\t wt\t tt\n");
 for(i=0;i<n;i++)
  printf("%d\t %d\t %d\n",bt[i],wt[i],tt[i]);
 printf("aw=%f\n,at=%f\n",aw,at);
}
```

1) Perform a trial run on the following example to check your program:

INPUT

```
   Enter number of processes
          3
   Enter burst time (execution time)
          12
          8
          20
```

OUTPUT : bt wt tt

```
              12 0 12
              8 12 20
              20 20 40

              aw=10.666670    /* Average waiting time */
              at=24.00000     /* Average turnaround time */
```

Exercise n°2

1) In this exercise you will be asked to implement the SJF scheduling algorithm and calculate the waiting time and turnaround time for each process in two versions: y
   a) non-preemptive SJF,
   b) preemptive SJF.

For the SJF scheduling algorithm, read the number of processes in the system, their CPU execution times. Rank all processes in order according to their execution time. There may be two processes in the queue with the same execution time, and the FCFS approach should then be applied to break the tie. Each process will be executed according to its runtime.

Version non préemptive de SJF

```
#include<stdio.h>
int i,j,bt[10],t,n,wt[10],tt[10],w1=0,t1=0;
float aw,at;
void main()
{
 printf("enter no. of processes:\n");
 scanf("%d",&n);
 printf("enter the burst time of processes:");
 for(i=0;i<n;i++)
  scanf("%d",&bt[i]);
 for(i=0;i<n;i++)
 {
  for(j=i;j<n;j++)
   if(bt[i]>bt[j])
   {
    t=bt[i];
    bt[i]=bt[j];
    bt[j]=t;
   }
 }
 for(i=0;i<n;i++)
   printf("%d",bt[i]);
 for(i=0;i<n;i++)
 {
  wt[0]=0;
  tt[0]=bt[0];
  wt[i+1]=bt[i]+wt[i];
  tt[i+1]=tt[i]+bt[i+1];
  w1=w1+wt[i];
  t1=t1+tt[i];
 }
 aw=w1/n;
 at=t1/n;
 printf("\nbt\t wt\t tt\n");
 for(i=0;i<n;i++)
    printf("%d\t %d\t %d\n",bt[i],wt[i],tt[i]);
 printf("aw=%f\n,at=%f\n",aw,at);
}
```

2) Perform a trial run on the following example to check your program:

INPUT:

Enter no of processes

    3

Enter burst time

    12
    8
    20

OUTPUT: bt wt tt

```
          12 8 20
          8  0 8
```

20 20 40

aw=9.33 at=22.64