SPOT EXERCISE

LAB 12

P.PRIYADHARSHINI

2019103562

26-04-2021

SRTF(SHORTEST REMAINING TIME FIRST):

```
[s2019103562@centos8-linux Mon Apr 26 11:16 AM lab12]$ vim srtf.c
[s2019103562@centos8-linux Mon Apr 26 11:18 AM lab12]$ gcc srtf.c -o srtf
[s2019103562@centos8-linux Mon Apr 26 11:18 AM lab12]$ ./srtf
Enter the number of processes:  3
Enter the arrival time for process 1:   0

Enter the burst time for process 1:     8
Enter the arrival time for process 2:   1

Enter the burst time for process 2:     2
Enter the arrival time for process 3:   4

Enter the burst time for process 3:     3

AVERAGE WAIITNG TIME:   0.333333
AVERAGE TURNAROUND TIME:        4.666667
```

```
[s2019103562@centos8-linux Mon Apr 26 11:20 AM lab12]$ cat srtf.c
#include<stdio.h>
int main(){
        int n,count=0,i,j,smallest,time;
        double avg=0,tt=0,end;
        printf("Enter the number of processes:\t");
        scanf("%d",&n);
        int at[10],bt[10],x[10];
        for(i=0;i<n;i++){
                printf("Enter the arrival time for process %d:\t",i+1);
                scanf("%d",&at[i]);
                printf("\nEnter the burst time for process %d:\t",i+1);
                scanf("%d",&bt[i]);
        }
        for(i=0;i<n;i++)
                x[i]=bt[i];
        bt[9]=9999;
        for(time=1;count!=n;time++){
                smallest=9;
                for(i=0;i<n;i++){
                        if(at[i]<=time && bt[i]<bt[smallest] && bt[i]>0){
                                smallest=i;
                                bt[smallest]--;
                                if(bt[smallest]==0){
                                        count++;
                                        end=time+1;
                                        avg=avg+end-at[smallest]-x[smallest];
                                        tt=tt+end-at[smallest];
                                }
                        }
                }
        }
                printf("\nAVERAGE WAIITNG TIME:\t%f",avg/n);
                printf("\nAVERAGE TURNAROUND TIME:\t%f\n",tt/n);
                return 0;
        }

[s2019103562@centos8-linux Mon Apr 26 11:20 AM lab12]$
```

PREEMPTIVE PRIORITY SCHEDULING ALGORITHM:

```
[s2019103562@centos8-linux Mon Apr 26 11:54 AM lab12]$ ./pripre
Enter the number of processes:
4
Enter the details for processes[A]:
Enter arrival time:     0

Enter the burst time:   19

Enter priority: 2
Enter the details for processes[B]:
Enter arrival time:     1

Enter the burst time:   12

Enter priority: 1
Enter the details for processes[C]:
Enter arrival time:     2

Enter the burst time:   15

Enter priority: 0
Enter the details for processes[D]:
Enter arrival time:     3

Enter the burst time:   25

Enter priority: 3
```

```
PROCESS          ARRIVAL TIME            BURST TIME              PRIORITY                WAITING TIME            TURNAROUNDTIME

A                       0                       19                      2                       1                       20

D                       3                       25                      3                       17                      42

B                       1                       12                      1                       44                      56

C                       2                       15                      0                       55                      70

 AVERAGE WAITING TIME:  29.250000
 AVERAGE TURNAROUND TIME:       47.000000
[s2019103562@centos8-linux Mon Apr 26 11:54 AM lab12]$ vim pripre.c
```

```
[s2019103562@centos8-linux Mon Apr 26 11:26 AM lab12]$ vim pripre.c
[s2019103562@centos8-linux Mon Apr 26 11:42 AM lab12]$ gcc pripre.c -o pripre
[s2019103562@centos8-linux Mon Apr 26 11:42 AM lab12]$ ./pripre
Enter the number of processes:
3
Enter the details for processes[A]:
Enter arrival time:     1

Enter the burst time:   23

Enter priority: 2
Enter the details for processes[B]:
Enter arrival time:     2

Enter the burst time:   54

Enter priority: 1
Enter the details for processes[C]:
Enter arrival time:     3

Enter the burst time:   12

Enter priority: 3

PROCESS          ARRIVAL TIME            BURST TIME              PRIORITY                WAITING TIME            TURNAROUNDTIME

A                       1                       23                      2                       1                       24

C                       3                       12                      3                       22                      34

B                       2                       54                      1                       35                      89

 AVERAGE WAITING TIME:  19.333334
 AVERAGE TURNAROUND TIME:       49.000000
[s2019103562@centos8-linux Mon Apr 26 11:43 AM lab12]$
```

```
[s2019103562@centos8-linux Mon Apr 26 11:43 AM lab12]$ cat pripre.c
#include<stdio.h>
struct process{
        char process_name;
        int arrival_time,burst_time,ct,waiting_time,turnaround_time,priority;
        int status;
}process_queue[10];
int limit;
void arrival_time_sorting(){
        struct process temp;
        int i,j;
        for(i=0;i<limit-1;i++){
                for(j=i+1;j<limit;j++){
                        if(process_queue[i].arrival_time>process_queue[j].arrival_time){
                                temp=process_queue[i];
                                process_queue[i]=process_queue[j];
                                process_queue[j]=temp;
                        }
                }
        }
}
int main(){
        int i,time=1,burst_time=0,largest;
        char c;
        float wait_time=0,turnaround_time=0,average_waiting_time,average_turnaround_time;
        printf("Enter the number of processes:\n");
        scanf("%d",&limit);
        for(i=0,c='A';i<limit;c++,i++){
                process_queue[i].process_name=c;
                printf("Enter the details for processes[%c]:\n",process_queue[i].process_name);
                printf("Enter arrival time:\t");
                scanf("%d",&process_queue[i].arrival_time);
```

```
                printf("\nEnter the burst time:\t");
                scanf("%d",&process_queue[i].burst_time);
                printf("\nEnter priority:\t");
                scanf("%d",&process_queue[i].priority);
                process_queue[i].status=0;
                burst_time=burst_time+process_queue[i].burst_time;
        }
        arrival_time_sorting();
        process_queue[9].priority=-9999;
        printf("\nPROCESS\t\t ARRIVAL TIME\t\t BURST TIME\t\t PRIORITY\t\t WAITING TIME\t\t TURNAROUNDTIME\n");
        for(time=process_queue[0].arrival_time;time<burst_time;){
                largest=9;
                for(i=0;i<limit;i++){
                        if(process_queue[i].arrival_time<=time && process_queue[i].status!=1 && process_queue[i].priority>process_queue[largest].priority){
                                largest=i;
                        }
                }
                time=time+process_queue[largest].burst_time;
                process_queue[largest].ct=time+1;
                process_queue[largest].waiting_time=process_queue[largest].ct-process_queue[largest].arrival_time-process_queue[largest].burst_time;
                process_queue[largest].turnaround_time=process_queue[largest].ct-process_queue[largest].arrival_time;
                process_queue[largest].status=1;
                wait_time=wait_time+process_queue[largest].waiting_time;
                turnaround_time=turnaround_time+process_queue[largest].turnaround_time;
                printf("\n%c\t\t\t %d\t\t\t %d\t\t\t %d\t\t\t %d\t\t\t %d\n",process_queue[largest].process_name,process_queue[largest].arrival_time,process_queue[large
st].burst_time,process_queue[largest].priority,process_queue[largest].waiting_time,process_queue[largest].turnaround_time);
        }
        average_waiting_time=wait_time/limit;
        average_turnaround_time=turnaround_time/limit;
        printf("\n AVERAGE WAITING TIME:\t%f",average_waiting_time);
        printf("\n AVERAGE TURNAROUND TIME:\t%f\n",average_turnaround_time);
        return 0;
        }

[s2019103562@centos8-linux Mon Apr 26 11:45 AM lab12]$
```