## **Operating System Lab Observation**

Mohith Jain 1BM22CS162

Lab 1:

```
Write a Cprogram to stimulate the following
non-pre-emptive CPU scheduling aloprithms to
# include (staio. h)
int main () ?
jut n, arrv-t[30], bor-t[30], com-t[30], tat[30],
        com-t[i] = temp + box.+[i];
```

Classostic. Date 21 111	classingte  Dole ald 14  Ogg 3			
O MAN OF THE PARTY	0			
else §	Outpot:			
else }				
tout (13- divit (13)	taster the number of protesses: 4			
lise & count (i] = arret (i] + box-t[i];  temp are t[i];	Frater the Aximal time:			
5373 4	P. 17 17 : 0			
to GT = court GT - onvetby	p-1d [2]: 1			
	Enter the Morrison time:  P.id[2]: 0  P.1d[2]: 1  P.1d[3]: 5			
temp = court [i];	P-Id[H] + 6 min [] make her			
	Blediens i Osi dai lad			
ang-tat-lit=tat [] + man + i	Entry the Borry Time:			
dara-wt-t += wart tay	0:01:1:12:02:00			
3 - s and good of a see quet trail	P.10[7] 2 P-10[5] 3			
	P-id (3) ; 3			
print C'ha Procus id It Amual time It Burritime	Prid [A]: H			
It compile time It Town trooped Time It				
Waiting line In I	Rocen ld Arrival time Bossi fine Coupie 1 AT hearty			
for (int 120) (Kur) 17th				
Wenting Pine In")  for (int 1 - Or ixerint) of the trid little year	P-100 0 2 2 2 0			
	ρ-rd1 1 2 4 3 1 ρ-rd2 5 3 3 3 0			
(E) (C) (C) (C) (C) (C)	P-1d 2 5 3 8 3 0			
bost (i), con + (i) tat (i), want + (i)	D-803 6 H 12 6 2			
16	To say ( that so the was been of mooned: I			
( " and every all vited at ) though	Acros bu Avand time = 2,5000			
ang-tat-lf-ni(minsingini)	Marios hosting the ton to Dat Dozo			
ang tot el = n;	College See See See See See See See See See			
· · · · · · · · · · · · · · · · · · ·	Horoge waiting time: 0.7000			
print ( In thereope twent time = 1 f , ang tal 1)	(Maryon Con ) Long			
prival ( In Average weeting time = 1. of , oug-wi-t);				
(tri in Loci tri) rat	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			
Netrono;				
3 (quot => [i] + xxx) (i	(" with read all are I'm " " rece			
	Plant to all deliberations			
(Class a gest Clitery	(1/2 1 1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/2 1/			
3	(13) + end & marin 3			
	THE RESIDENCE OF THE PARTY OF T			

```
#include <stdio.h>
void swap(int *a, int *b) {
   int temp = *a;
   *a = *b;
    *b = temp;
}
void sort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
               swap(&arr[j], &arr[j + 1]);
       }
    }
int main() {
   int n, arrv t[30], bur t[30], com t[30], wait t[30], tat[30], temp = 0;
    float avg_wt_t = 0, avg_tat_t = 0;
    printf("\nEnter the number of processes:");
    scanf("%d", &n);
    printf("\nEnter the arrival time:");
    for (int i = 0; i < n; i++) {
        printf("P_id[%d]:", i + 1);
        scanf("%d", &arrv_t[i]);
    }
    printf("\nEnter the burst time:");
    for (int i = 0; i < n; i++) {
       printf("P_id[%d]:", i + 1);
       scanf("%d", &bur t[i]);
    sort(bur_t, n); // Sort burst times
    for (int i = 0; i < n; i++) {
        if (arrv t[i] <= temp) {</pre>
            com t[i] = temp + bur t[i];
        } else {
            com t[i] = arrv t[i] + bur t[i];
            temp = arrv t[i];
        tat[i] = com_t[i] - arrv_t[i];
        wait t[i] = tat[i] - bur t[i];
        temp = com_t[i];
        avg tat t += tat[i];
        avg_wt_t += wait_t[i];
    }
```

```
printf("\nProcess id \t Arrival time \t Burst time \t Compile time\t TurnAround time \t
Waiting time\n");
    for (int i = 0; i < n; i++) {
        printf("\nP_id%d \t\t\t %d \t\t %d \t\t %d \t\t %d \t\t %d \t\t %d\n", i, arrv_t[i],
bur_t[i], com_t[i], tat[i], wait_t[i]);
    }
    avg_tat_t /= n;
    avg_wt_t /= n;
    printf("\n Average turnAround time = %f", avg_tat_t);
    printf("\n Average waiting time = %f", avg_wt_t);
    return 0;
}</pre>
```

```
Enter the number of processes:4
 Enter the arrival time:P_id[1]:0
 P_id[2]:1
P_id[3]:5
 P_id[4]:6
 Enter the burst time:P_id[1]:2
 P_id[2]:2
P_id[3]:3
 P_id[4]:4
 Process id
                   Arrival time
                                    Burst time
                                                     Compile time
                                                                      TurnAround time Waiting time
                                                             2
 P_id0
                            0
                                             2
                                                                              2
 P_id1
                                                             4
                                                                              3
 P_id2
                                             3
                                                                              3
                                                                                       0
 P_id3
                            6
                                             4
                                                             12
                                                                                       2
  Average turnAround time = 3.500000
  Average waiting time = 0.750000%
o mohithjain@Mohiths-MacBook-Air Downloads %
```

## Priority Scheduling Non-preemptive

```
#include<stdio.h>
void swap(int *a, int *b)
    int temp = *a;
    *a= *b;
    *b = temp;
}
void sort(int arr[],int prior[],int n)
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (prior[j] > prior[j + 1]) {
                swap(&prior[j], &prior[j + 1]);
                swap(&arr[j], &arr[j + 1]);
            }
        }
    }
}
int main()
int n,prior[30],burt[30],tat[30],waitt[30],temp;
float avg_waitt=0,avg_tat=0;
printf("\nEnter the number of processes:");
 scanf("%d",&n);
 for(int i=0;i<n;i++)
     printf("\nEnter the burst time & Priority of process %d:",i);
      scanf("%d %d",&burt[i],&prior[i]);
 sort(burt,prior,n);
  for (int i = 0; i < n; i++) {
        waitt[i] = temp;
       tat[i] = waitt[i] + burt[i];
        temp += burt[i];
        avg tat += tat[i];
        avg_waitt += waitt[i];
    }
    printf("\nProcess id \t Burst time \t Priority \t TurnAround time \t Waiting time\n");
    for (int i = 0; i < n; i++) {
        printf("\nP_id%d \t\t\ %d \t\t\ %d \t\t\ %d\n", i, burt[i], prior[i],
tat[i], waitt[i]);
   }
    avg_tat /= n;
```

```
avg_waitt /= n;
printf("\n Average turnAround time = %f", avg_tat);
printf("\n Average waiting time = %f", avg_waitt);
return 0;
```

Enter the number of processes:5						
Enter the burst time & Priority of process 0:10 3						
Enter the burst time & Priority of process 1:1 1						
Enter the burst time & Priority of process 2:2 4						
Enter the burst time & Priority of process 3:1 5						
Enter the burst time & Priority of process 4:5 2						
Process id	Burst time	Priority	TurnAround	time Waiting	time	
P_id0	1	1	1		0	
P_id1	5	2	6		1	
P_id2	10	3	16		6	
P_id3	2	4	18		16	
P_id4	1	5	19		18	
Average turnAround time = 12.000000  Average waiting time = 8.2000002  o mohithjain@Mohiths-MacBook-Air Downloads %						

## **Priority Scheduling Preemptive**

```
#include <stdio.h>
#define MAX 30
void sort by priority(int n, int bur t[], int pri[], int id[]) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (pri[i] > pri[j]) {
                temp = pri[i];
                pri[i] = pri[j];
                pri[j] = temp;
                temp = bur_t[i];
                bur t[i] = bur t[j];
                bur_t[j] = temp;
                temp = id[i];
                id[i] = id[j];
                id[j] = temp;
        }
    }
int main() {
   int n;
   int bur t[MAX], rem bur t[MAX], pri[MAX], id[MAX];
   int tat[MAX], wait_t[MAX];
   float avg_waiting_time = 0, avg_turnaround_time = 0;
   printf("Enter the number of processes: ");
    scanf("%d", &n);
    for (int i = 0; i < n; i++) {
        id[i] = i + 1;
        printf("Enter burst time of process P%d: ", id[i]);
        scanf("%d", &bur t[i]);
        rem bur t[i] = bur t[i];
        printf("Enter priority of process P%d: ", id[i]);
        scanf("%d", &pri[i]); }
    sort by priority(n, bur t, pri, id);
    int current time = 0, completed = 0;
    while (completed != n) {
        int idx = -1;
        int highest_priority = 10000; // assuming a large number as the initial highest
priority
        for (int i = 0; i < n; i++) {
            if (rem bur t[i] > 0) {
                if (pri[i] < highest priority) {</pre>
                    highest priority = pri[i];
                    idx = i;
                if (pri[i] == highest priority) {
                    if (id[i] < id[idx]) {</pre>
                        highest priority = pri[i];
                        idx = i;
                    }
```

```
}
        if (idx != -1) {
            current time++;
            rem bur t[idx]--;
            if (rem_bur_t[idx] == 0) {
                completed++;
                tat[idx] = current_time;
                wait t[idx] = tat[idx] - bur t[idx];
                avg_waiting_time += wait_t[idx];
                avg_turnaround_time += tat[idx];
        } else {
            current_time++;
    printf("\nProcess ID\tBurst Time\tPriority\tTurnaround Time\tWaiting Time\n");
    for (int i = 0; i < n; i++) {
        printf("P%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t, id[i], bur t[i], pri[i], tat[i],
wait t[i]);
    }
    avg_waiting_time /= n;
    avg turnaround time /= n;
    printf("\nAverage Turnaround Time = %.2f", avg turnaround time);
    printf("\nAverage Waiting Time = %.2f\n", avg waiting time);
    return 0;
}
```

```
Enter the number of processes: 5
Enter burst time of process P1: 10
Enter priority of process P1: 3
Enter burst time of process P2: 1
Enter priority of process P2: 1
Enter burst time of process P3: 2
Enter priority of process P3: 4
Enter burst time of process P4: 1
Enter priority of process P4: 5
Enter burst time of process P5: 5
Enter priority of process P5: 2
Process ID
                Burst Time
                                Priority
                                                 Turnaround Time Waiting Time
P2
                                                                 6
                                2
P5
                5
                                                 11
Ρ1
                                3
                10
                                                 13
Р3
                                4
                                                 14
                                                                 12
                1
                                                 19
Average Turnaround Time = 13.40
Average Waiting Time = 9.60
mohithjain@Mohiths-MacBook-Air Lab %
```

## Round Robin

```
#include <stdio.h>
int main() {
   int n, i, tq, time = 0;
   int bur t[30], rem bur t[30], wait t[30], tat[30];
   float avg wt t = 0, avg tat t = 0;
   printf("\nEnter the number of processes: ");
    scanf("%d", &n);
    printf("\nEnter the burst time for each process:\n");
    for (i = 0; i < n; i++) {
       printf("Burst time of process P%d: ", i + 1);
       scanf("%d", &bur_t[i]);
       rem bur t[i] = bur t[i];
   printf("\nEnter the time quantum: ");
    scanf("%d", &tq);
   int done = 0;
    while (done != n) {
       for (i = 0; i < n; i++) {
           if (rem bur t[i] > 0) {
                if (rem bur t[i] > tq) {
                   time += tq;
                    rem bur t[i] -= tq;
                } else {
                   time += rem bur t[i];
                    wait t[i] = time - bur t[i];
                    tat[i] = time;
                    rem_bur_t[i] = 0;
                    done++;
               }
           }
       }
    printf("\nProcess ID\tBurst Time\tTurnaround Time\tWaiting Time\n");
    for (i = 0; i < n; i++) {
        avg wt t += wait t[i];
        avg tat t += tat[i];
        printf("P%d\t\t%d\t\t\d\n", i + 1, bur t[i], tat[i], wait t[i]);
    avg wt t /= n;
    avg tat t /= n;
    printf("\nAverage Turnaround Time = %.2f", avg_tat_t);
    printf("\nAverage Waiting Time = %.2f\n", avg wt t);
    return 0;
```

```
Enter the number of processes: 3
 Enter the burst time for each process:
 Burst time of process P1: 24
 Burst time of process P2: 3
 Burst time of process P3: 3
 Enter the time quantum: 3
                                 Turnaround Time Waiting Time
 Process ID
                 Burst Time
 P1
                 24
                                 30
                                                 3
 P2
                 3
                                 6
                                                 6
 Р3
                 3
                                 9
 Average Turnaround Time = 15.00
Average Waiting Time = 5.00
⊃ mohithjain@Mohiths-MacBook-Air T % ■
```