Bansilal Ramnath Agarwal Charitable Trust'
Vishwakarma Institute of Technology
(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Operating System Lab

Assignment No: 3

Name : Gourav Suram

Roll No : 39

Prn No : 12220032

First Come First Serve Scheduling.

Program

```c
#include <stdio.h>

int main(){

  int n = 4;
  int at[] = {0, 1, 3, 3};
  int bt[] = {1, 9, 1, 9};

  int ft[n];
  for(int i=0; i<n; i++){
    ft[i] = ft[i-1]+bt[i];
  }

  int tt[15];
  for(int i=0; i<n;i++){
    tt[i] = ft[i] - at[i];
  }

  int wt[15];
  for(int i=0;i<n;i++){
    wt[i] = tt[i] - bt[i];
  }

  printf("\nAT\t\tBT\t\tFT\t\tTAT\t\tWT\n");
  for(int i=0; i<n; i++){
    printf("%d\t\t%d\t\t%d\t\t%d\t\t%d\n", i, bt[i], ft[i], tt[i], wt[i]);
  }
  float avg_wt = 0.0, avg_tt = 0.0;
  for (int i = 0; i < n; i++) {
    avg_wt += wt[i];
    avg_tt += tt[i];
  }
  avg_wt /= n;
  avg_tt /= n;

  printf("\nAverage WT:  %0.2f", avg_wt);
  printf("\nAverage TAT: %0.2f\n", avg_tt);

  return 0;
}
```

Output:

```
~/vit-comp/Module-4/Operating-System/Assignment-3 on main +3
❯ gcc FCFS.c -o Binary/FCFS && ./Binary/FCFS

AT              BT              FT              TAT             WT
0               1               1               1               0
1               9               10              9               0
2               1               11              8               7
3               9               20              17              8

Average WT:  3.75
Average TAT: 8.75
```

Shortest Job First Preemptive Scheduling.

Program:

```python
n = 4
bt = [[1, 0, 0], [9, 1, 1], [1, 3, 2], [9, 3, 3]]
at = [0, 1, 3, 3, 0]
abt = [1, 9, 1, 9, 0]
sumbt = 0
i = 0
ll = []
for i in range(0, sum(abt)):
    l = [j for j in bt  if j[1] <= i]
    l.sort(key=lambda x: x[0])
    bt[bt.index(l[0])][0] -= 1
    for k in bt:
        if k[0] == 0:
            t = bt.pop(bt.index(k))
            ll.append([k, i + 1])
ct = [0] * (n + 1)
tat = [0] * (n + 1)
wt = [0] * (n + 1)
for i in ll:
    ct[i[0][2]] = i[1]

for i in range(len(ct)):
    tat[i] = ct[i] - at[i]
    wt[i] = tat[i] - abt[i]
ct.pop(-1)
wt.pop(-1)
tat.pop(-1)
abt.pop(-1)
at.pop(-1)
print('AT\tBT\tFT\tTAT\tWT')
for i in range(len(ct)):
    print("{}\t{}\t{}\t{}\t{}\n".format(at[i],abt[i], ct[i], tat[i], wt[i]))
print('Average Waiting Time = ', sum(wt)/len(wt))
print('Average Turnaround Time = ', sum(tat)/len(tat))

```

Output:

```
~/vit-comp/Module-4/Operating-System/Assignment-3 on main +3
❯ python3 SJF-P.py
AT      BT      FT      TAT     WT
0       1       1       1       0
1       9       11      10      1
3       1       4       1       0
3       9       20      17      8

Average Waiting Time =  2.25
Average Turnaround Time =  7.25
```

Shortest Job First NonPreemptive Scheduling.

Program:

```c
#include <stdio.h>

int main(){
    int at[] = {0, 1, 3, 3};
    int bt[] = {1, 9, 1, 9};
    int n = sizeof(at)/ sizeof(at[0]);

    int ft[n], tat[n], wt[n];
    int total_wt = 0, total_tat = 0;

    int i, j, min, temp;
    for (i = 0; i < n; i++){
        min = i;
        for (j = i + 1; j < n; j++){
            if (at[j] < at[min]){
                min = j;
            }
        }
        temp = at[i];
        at[i] = at[min];
        at[min] = temp;

        temp = bt[i];
        bt[i] = bt[min];
        bt[min] = temp;
    }

    ft[0] = at[0] + bt[0];
    tat[0] = ft[0] - at[0];
    wt[0] = tat[0] - bt[0];

    for (i = 1; i < n; i++){
        ft[i] = ft[i - 1] + bt[i];
        tat[i] = ft[i] - at[i];
        wt[i] = tat[i] - bt[i];
    }

    printf("AT\t BT\t FT\t TAT\t WT");
    for (i = 0; i < n; i++){
        printf("\n%d\t %d\t %d\t %d\t %d", at[i], bt[i], ft[i], tat[i], wt[i]);
        total_wt += wt[i];
        total_tat += tat[i];
    }

    printf("\n\nAverage WT:  %0.2f", (total_wt * 1.0 / n));
    printf("\nAverage TAT: %0.2f\n", (total_tat * 1.0 / n));

    return 0;
}
```

Output:

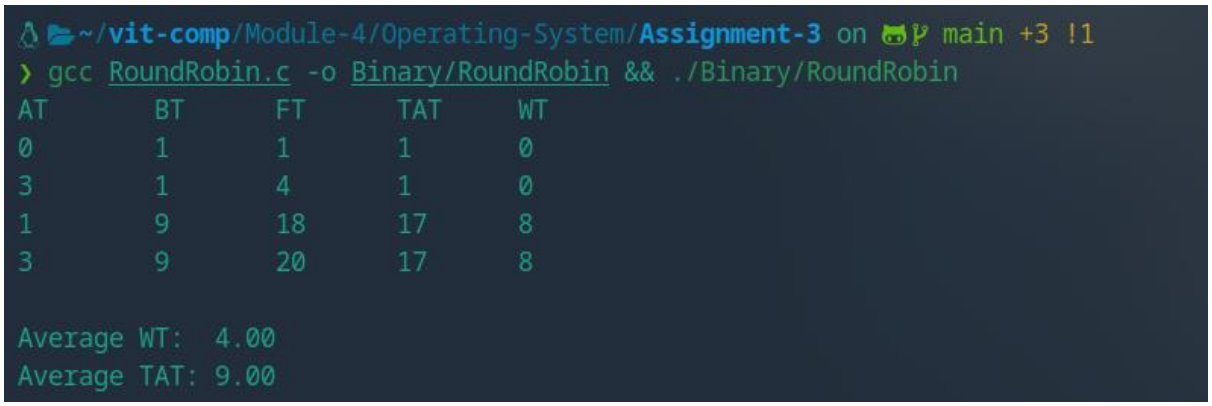
```
~/vit-comp/Module-4/Operating-System/Assignment-3 on  main +3 !1
❯ gcc SJF-NP.c -o Binary/SJF-NP && ./Binary/SJF-NP
AT      BT      FT      TAT     WT
0       1       1       1       0
1       9       10      9       0
3       1       11      8       7
3       9       20      17      8

Average WT:  3.75
Average TAT: 8.75
```

Round Robin Scheduling.

Program:

```c
#include <stdio.h>

int main()
{
    int i, n, time, remain, flag = 0, time_quantum = 1;
    int wait_time = 0, turnaround_time = 0, finish_time[10], rt[10];
    int at[] = {0,1,3,3};
    int bt[] = {1,9,1,9};
    n = sizeof at / sizeof at[0];
    remain = n;

    for (int i = 0; i < n; i++){
        rt[i] = bt[i];
    }

    printf("AT\t BT\t FT\t TAT\t WT");
    for (time = 0, i = 0; remain != 0;){
        if (rt[i] <= time_quantum && rt[i] > 0){
            time += rt[i];
            rt[i] = 0;
            flag = 1;
        }else if (rt[i] > 0){
            rt[i] -= time_quantum;
            time += time_quantum;
        }
        if (rt[i] == 0 && flag == 1){
            remain--;
            finish_time[i] = time;
            printf("\n%d\t %d\t %d\t %d\t %d", at[i],bt[i], finish_time[i],
finish_time[i] - at[i], finish_time[i] - at[i] - bt[i]);

            wait_time += finish_time[i] - at[i] - bt[i];
            turnaround_time += finish_time[i] - at[i];
            flag = 0;
        }
        if (i == n - 1){
            i = 0;
        } else if (at[i + 1] <= time){
            i++;
        } else{ i = 0; }
    }

    printf("\n\nAverage WT:  %0.2f", (wait_time * 1.0 / n));
    printf("\nAverage TAT: %0.2f\n", (turnaround_time * 1.0 / n));

    return 0;
}

```

Output:


~/vit-comp/Module-4/Operating-System/Assignment-3 on main +3 !1
gcc RoundRobin.c -o Binary/RoundRobin && ./Binary/RoundRobin
AT BT FT TAT WT
0 1 1 1 0
3 1 4 1 0
1 9 18 17 8
3 9 20 17 8
Average WT: 4.00
Average TAT: 9.00

Priority Based Scheduling - Non preemptive.

Program:

```c
#include<stdio.h>
#define MAX 9999;

struct proc{
    int no,at,bt,ct,wt,tat,pri,status;
};
struct proc read(int i){
    struct proc p;
    printf("\nProcess No: %d\n",i);
    p.no=i;
    printf("Enter Arrival Time: ");
    scanf("%d",&p.at);
    printf("Enter Burst Time: ");
    scanf("%d",&p.bt);
    printf("Enter Priority: ");
    scanf("%d",&p.pri);
    p.status=0;
    return p;
}

int main(){

    int n,s,ct=0,remaining;
    struct proc p[10],temp;
    float avgtat=0,avgwt=0;

    printf("Enter Number of Processes: ");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        p[i]=read(i+1);
    for(int i=0;i<n-1;i++)
        for(int j=0;j<n-i-1;j++)
            if(p[j].at>p[j+1].at)
            {
            temp=p[j];
            p[j]=p[j+1];
            p[j+1]=temp;
            }
    p[9].pri=MAX;
    remaining=n;
    printf("\nProcessNo\tAT\tBT\tPri\tCT\tTAT\tWT\tRT\n");
    for(ct=p[0].at;remaining!=0;)
    {
        s=9;
        for(int i=0;i<n;i++)
            if(p[i].at<=ct && p[i].status!=1 && p[i].pri<p[s].pri)
                s=i;
        p[s].ct=ct=ct+p[s].bt;
        p[s].tat=p[s].ct-p[s].at;
        avgtat+=p[s].tat;
        p[s].wt=p[s].tat-p[s].bt;
        avgwt+=p[s].wt;
        p[s].status=1;
        remaining--;

   printf("P%d\t\t%d\t%d\t%d\t%d\t%d\t%d\t%d\n",p[s].no,p[s].at,p[s].bt,p[s].pri,p[s].ct,p[s].
  tat,p[s].wt,p[s].wt);
    }
    avgtat/=n,avgwt/=n;
    printf("\nAverage TurnAroundTime=%0.2f\nAverage WaitingTime=%0.2f\n",avgtat,avgwt);

    return 0;
}
```

Output:

```
~/vit-comp/Module-4/Operating-System/Assignment-3 on main +3 !1
❯ gcc Priority-NP.c -o Binary/Priority-NP && ./Binary/Priority-NP
Enter Number of Processes: 4

Process No: 1
Enter Arrival Time: 0
Enter Burst Time: 1
Enter Priority: 3

Process No: 2
Enter Arrival Time: 1
Enter Burst Time: 9
Enter Priority: 3

Process No: 3
Enter Arrival Time: 3
Enter Burst Time: 1
Enter Priority: 2

Process No: 4
Enter Arrival Time: 3
Enter Burst Time: 9
Enter Priority: 1

ProcessNo       AT      BT      Pri     CT      TAT     WT      RT
P1              0       1       3       1       1       0       0
P2              1       9       3       10      9       0       0
P4              3       9       1       19      16      7       7
P3              3       1       2       20      17      16      16

Average TurnAroundTime=10.75
Average WaitingTime=5.75
```

Priority Based Scheduling - Preemptive.

Program:

```c
#include<stdio.h>
#define MAX 9999;

struct proc{
    int no,at,bt,rt,ct,wt,tat,pri,temp;
};

struct proc read(int i){
    struct proc p;
    printf("\nProcess No: %d\n",i);
    p.no=i;
    printf("Enter Arrival Time: ");
    scanf("%d",&p.at);
    printf("Enter Burst Time: ");
    scanf("%d",&p.bt);
    p.rt=p.bt;
    printf("Enter Priority: ");
    scanf("%d",&p.pri);
    p.temp=p.pri;
    return p;
}

int main(){
    int i,n,c,remaining,min_val,min_index;
    struct proc p[10],temp;
    float avgtat=0,avgwt=0;

    printf("Enter Number of Processes: ");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        p[i]=read(i+1);
    remaining=n;
    for(int i=0;i<n-1;i++)
        for(int j=0;j<n-i-1;j++)
            if(p[j].at>p[j+1].at){
            temp=p[j];
            p[j]=p[j+1];
            p[j+1]=temp;
            }
    min_val=p[0].temp,min_index=0;
    for(int j=0;j<n&&p[j].at<=p[0].at;j++)
        if(p[j].temp<min_val)
            min_val=p[j].temp,min_index=j;
    i=min_index;
    c=p[i].ct=p[i].at+1;
    p[i].rt--;
    if(p[i].rt==0){
        p[i].temp=MAX;
        remaining--;
    }
    while(remaining>0){
        min_val=p[0].temp,min_index=0;
        for(int j=0;j<n&&p[j].at<=c;j++)
            if(p[j].temp<min_val)
                min_val=p[j].temp,min_index=j;
        i=min_index;
        p[i].ct=c=c+1;
        p[i].rt--;
        if(p[i].rt==0)
        {
            p[i].temp=MAX;
            remaining--;
        }
    }
        printf("\nProcessNo\tAT\tBT\tPri\tCT\tTAT\tWT\n");
    for(int i=0;i<n;i++) {
        p[i].tat=p[i].ct-p[i].at;
        avgtat+=p[i].tat;
        p[i].wt=p[i].tat-p[i].bt;
        avgwt+=p[i].wt;

 printf("P%d\t\t%d\t%d\t%d\t%d\t%d\t%d\n",p[i].no,p[i].at,p[i].bt,p[i].pri,p[i].ct,p[i].tat,
p[i].wt);
    }
    avgtat/=n,avgwt/=n;
    printf("\nAverage TurnAroundTime=%0.2f\nAverage WaitingTime=%0.2f\n",avgtat,avgwt);

    return 0;
}
```

Output:

```
~/vit-comp/Module-4/Operating-System/Assignment-3 on main +3 !1
❯ gcc Priority-P.c -o Binary/Priority-P && ./Binary/Priority-P
Enter Number of Processes: 4

Process No: 1
Enter Arrival Time: 0
Enter Burst Time: 1
Enter Priority: 3

Process No: 2
Enter Arrival Time: 1
Enter Burst Time: 9
Enter Priority: 3

Process No: 3
Enter Arrival Time: 3
Enter Burst Time: 1
Enter Priority: 2

Process No: 4
Enter Arrival Time: 3
Enter Burst Time: 9
Enter Priority: 1

ProcessNo       AT      BT      Pri     CT      TAT     WT
P1              0       1       3       1       1       0
P2              1       9       3       20      19      10
P3              3       1       2       13      10      9
P4              3       9       1       12      9       0

Average TurnAroundTime=9.75
Average WaitingTime=4.75
```