

[illegible]

```

pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ gedit fcfs5095.c
^C
pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ gcc fcfs5095.c -lpthread
pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ ./a.out
FCFS
Number of Process:6
id at bt
4 0 10
5 1 6
1 2 4
6 4 2
2 5 1
3 6 3
Process Id      Arrival Time    Bust Time      Completion Time  Turn around Time  Waiting Time
P4              0              10             10              10              0
P5              1              6              16              15              9
P1              2              4              20              18              14
P6              4              2              22              18              16
P2              5              1              23              18              17
P3              6              3              26              20              17
Average Waiting time:12.167
Average TAT :16.500
pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations/lab 6$

```

2. SJF NON-PRE-EMPTIVE

```

pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ cat sjfnonpre5095.c
#include<stdio.h>
#include<pthread.h>
int swap(int* i, int* j)
{
    int temp = *i;
    *i = *j;
    *j = temp;
}
void ctf(int ct[],int bt[],int at[],int pid[],int n)
{
    int min=1000,pos;
    ct[0]=at[0] + bt[0];
    for(int i=1; i<n; i++)
    {
        for(int j=i; j<n; j++)
        {
            if(at[j]<=ct[i-1])
            {
                if(bt[j]<min)
                {
                    min=bt[j];
                    pos=j;
                }
            }
        }
        swap(&pid[i], &pid[pos]);
        swap(&at[i], &at[pos]);
        swap(&bt[i], &bt[pos]);
        min=1000;
        ct[i]=ct[i-1]+bt[i];
    }
}
void tatf(int tat[],int ct[],int at[],int n)
{
    for(int i=0;i<n;i++)
        tat[i]=ct[i]-at[i];
}
void wtf(int wt[],int tat[],int bt[],int n)
{
    for(int i=0;i<n;i++)
        wt[i]=tat[i]-bt[i];
}
void * sjf()
{
    printf("SJF\n");
    int n,tp1,tp2,tp3;
    float wta=0,tata=0;
    printf("Number of Process:");
    scanf("%d",&n);
    int pid[n],at[n],bt[n],ct[n],tat[n],wt[n];
    printf("id at bt\n");
}

```

```

pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations/lab 6
printf("id at bt\n");
for(int i=0;i<n;i++)
{
    scanf("%d %d", &pid[i], &at[i], &bt[i]);
}
for(int i=0;i<n;i++)
{
    for(int j=0;j<n-1-i;j++)
    {
        if(at[j]>at[j+1])
        {
            swap(&at[j+1], &at[j]);
            swap(&pid[j+1], &pid[j]);
            swap(&bt[j+1], &bt[j]);
        }
        if(at[j]==at[j+1])
        {
            if(bt[j]>bt[j+1])
            {
                swap(&at[j+1], &at[j]);
                swap(&pid[j+1], &pid[j]);
                swap(&bt[j+1], &bt[j]);
            }
        }
    }
}
ctf(ct, bt, at, pid, n);
tatf(at, ct, at, n);
wtf(wt, tat, bt, n);
printf("Process Id\tArrival Time\tBurst Time\tCompletion Time\tTurn around Time\tWaiting Time\n");
for(int i=0;i<n;i++)
{
    tat[i]=ct[i]-at[i];
    tata=tat[i];
    wt[i]=tat[i]-bt[i];
    wta=wt[i];
    printf(" %d\t%d\t%d\t%d\t%d\t%d\t%d\n", pid[i], at[i], bt[i], ct[i], tat[i], wt[i]);
}
printf("Average Waiting time:%.3f\n", wta/n);
printf("Average TAT :%.3f\n", tata/n);
}

int main()
{
    pthread_t ti;
    pthread_create(&ti, NULL, (void*)sjf, NULL);
    pthread_join(ti, NULL);
    return 0;
}
pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ ^C
pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ _

```

```

pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations/lab 6
pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ gcc sjfnonpre5095.c -lpthread
pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ ./a.out
SJF
Number of Process:5
id at bt
1 4 7
2 3 9
3 0 10
4 6 13
5 5 11
Process Id      Arrival Time    Burst Time      Completion Time  Turn around Time  Waiting Time
P3              0              10              10              10              0
P1              4              7              17              13              6
P2              3              9              26              23              14
P5              5              11              37              32              21
P4              6              13              50              44              31
Average Waiting time:14.400
Average TAT :24.400
pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations/lab 6$

```

3. ROUND ROBIN(NON-PRE-EMPTIVE)

```

pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_obse...
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ cat rr5095.c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
//ROUND ROBIN
int main(){
    int i, NOP, sum=0, count=0, y, quant, wt=0, tat=0, at[10], bt[10], temp[10];
    float avg_wt, avg_tat;
    printf("Total number of process: ");
    scanf("%d", &NOP);
    y = NOP;
    for(i=0; i<NOP; i++){
        printf("\nEnter the Arrival and Burst time of the Process[%d]\n", i+1);
        printf("Arrival time is:");
        scanf("%d", &at[i]);
        printf("Burst time is:");
        scanf("%d", &bt[i]);
        temp[i] = bt[i];
    }
    printf("Enter the Time Quantum for the process:");
    scanf("%d", &quant);
    printf("\nProcess No \t\t Burst Time \t\t TAT \t\t Waiting Time");
    for(sum=0, i = 0; y!=0; ){
        if(temp[i] <= quant && temp[i] > 0){
            sum = sum + temp[i];
            temp[i] = 0;
            count=1;
        }
        else if(temp[i] > 0){
            temp[i] = temp[i] - quant;
            sum = sum + quant;
        }
        if(temp[i]==0 && count==1){
            y--;
            printf("\nP%d \t\t %d\t\t\t\t\t %d\t\t\t\t\t %d", i+1, bt[i], sum-at[i], sum-at[i]-bt[i]);
            wt = wt+sum-at[i]-bt[i];
            tat = tat+sum-at[i];
            count =0;
        }
        if(i==NOP-1){
            i=0;
        }
        else if(at[i+1]<=sum){
            i++;
        }
        else{
            i=0;
        }
    }
    avg_wt = wt * 1.0/NOP;
    avg_tat = tat * 1.0/NOP;
    printf("\n Average Turn Around Time: \t%f", avg_wt);
    printf("\n Average Waiting Time: \t%f", avg_tat);
    return 0;
}
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$

```

```

pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations/lab 6
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ gedit rr5095.c
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ gcc rr5095.c
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ ./a.out
Total number of process: 4

Enter the Arrival and Burst time of the Process[1]
Arrival time is:0
Burst time is:4

Enter the Arrival and Burst time of the Process[2]
Arrival time is:1
Burst time is:3

Enter the Arrival and Burst time of the Process[3]
Arrival time is:2
Burst time is:2

Enter the Arrival and Burst time of the Process[4]
Arrival time is:3
Burst time is:5
Enter the Time Quantum for the process:2

Process No      Burst Time      TAT      Waiting Time
P3              2              4          2
P1              4             10          6
P2              3             10          7
P4              5             11          6
Average Turn Around Time:      5.250000
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$

```

4. PRIORITY NON-PRE-EMPTIVE

```
pv@pv-Vostro-5402: /media/pv/...
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ cat prioritynon5095.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("<--Smallest Priority First Scheduling Algorithm (Non-Preemptive)-->\n");
    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\tTRT\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+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%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=%f\nAverage WaitingTime=%f",avgtat,avgwt);
    return 0;
}
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$
```

```

pv@pv-Vostro-5402: /media/pv/Personal/...
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ gcc prioritynon5095.c
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ ./a.out
<--Smallest Priority First Scheduling Algorithm (Non-Preemptive)-->
Enter Number of Processes: 4

Process No: 1
Enter Arrival Time: 1
Enter Burst Time: 5
Enter Priority: 3

Process No: 2
Enter Arrival Time: 3
Enter Burst Time: 4
Enter Priority: 5

Process No: 3
Enter Arrival Time: 4
Enter Burst Time: 7
Enter Priority: 2

Process No: 4
Enter Arrival Time: 2
Enter Burst Time: 6
Enter Priority: 1

ProcessNo    AT    BT    Pri    CT    TAT    WT    RT
P1           1     5     3     6     5     0     0
P4           2     6     1    12    10     4     4
P3           4     7     2    19    15     8     8
P2           3     4     5    23    20    16    16

Average TurnAroundTime=12.500000
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$

```

5. SJF PRE-EMPTIVE:

```

pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_o...
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ gedit sjfpre5095.c
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ gcc sjfpre5095.c
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ ./a.out
<--SRTF Scheduling Algorithm (Preemptive)-->
Enter Number of Processes: 5

Process No: 1
Enter Arrival Time: 2
Enter Burst Time: 6

Process No: 2
Enter Arrival Time: 5
Enter Burst Time: 2

Process No: 3
Enter Arrival Time: 1
Enter Burst Time: 8

Process No: 4
Enter Arrival Time: 0
Enter Burst Time: 3

Process No: 5
Enter Arrival Time: 4
Enter Burst Time: 4

Process    AT    BT    CT    TAT    WT
P4         0     3     3     3     0
P2         5     2     7     2     0
P5         4     4    10     6     2
P1         2     6    15    13     7
P3         1     8    23    22    14

Average TurnAroundTime=9.200000
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$

```

```
pv@pv-Vostro-5402: /media/pv/...
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ cat sjfpre5095.c
#include<stdio.h>
#define MAX 9999
struct proc
{
    int no,at,bt,rt,ct,tat,wt;
};
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;
    return p;
}
int main()
{
    struct proc p[10],temp;
    float avgtat=0,avgwt=0;
    int n,s,remain=0,time;
    printf("<--SRTF Scheduling Algorithm (Preemptive)-->\n");
    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;
            }
    printf("\nProcess\t\tat\tbt\tct\tTAT\tWT\n");
    p[9].rt=MAX;
    for(time=0;remain!=n;time++)
    {
        s=9;
        for(int i=0;i<n;i++)
            if(p[i].at<=time&& p[i].rt<p[s].rt&&p[i].rt>0)
                s=i;
        p[s].rt--;
        if(p[s].rt==0)
        {
            remain++;
            p[s].ct=time+1;
            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;
            printf("P%d\t\t%d\t%d\t%d\t%d\t%d\n",p[s].no,p[s].at,p[s].bt,p[s].ct,p[s].tat,p[s].wt);
        }
    }
    avgtat/=n,avgwt/=n;
    printf("\nAverage TurnAroundTime=%f\nAverage WaitingTime=%f",avgtat,avgwt);
}
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$
```

6. PRIORITY PRE-EMPTIVE:

```

pv@pv-Vostro-5402: /me...
pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations/Lab 6$ cat pbsper5095.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("<--Smallest Priority First Scheduling Algorithm (Preemptive)-->\n");
    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&&min_val<p[j].temp;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&&min_val<p[j].temp;j++)
            if(p[j].temp<min_val)
                min_val=p[j].temp,min_index=j;
        i=min_index;
        p[i].ct=c;
        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("%d\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=%f\nAverage WaitingTime=%f",avgtat,avgwt);
    return 0;
}
pv@pv-Vostro-5402: /media/pv/Personal/Code/C C++/os_lab_observations/Lab 6$

```



```
pv@pv-Vostro-5402: /media/pv/...  
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ gcc pbsper5095.c  
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$ ./a.out  
<--Smallest Priority First Scheduling Algorithm (Preemptive)-->  
Enter Number of Processes: 4  
  
Process No: 1  
Enter Arrival Time: 1  
Enter Burst Time: 5  
Enter Priority: 3  
  
Process No: 2  
Enter Arrival Time: 3  
Enter Burst Time: 4  
Enter Priority: 5  
  
Process No: 3  
Enter Arrival Time: 4  
Enter Burst Time: 7  
Enter Priority: 2  
  
Process No: 4  
Enter Arrival Time: 2  
Enter Burst Time: 6  
Enter Priority: 1  
  
ProcessNo    AT    BT    Pri    CT    TAT    WT  
P1           1     5     3     19    18    13  
P4           2     6     1     8     6     0  
P2           3     4     5    23    20    16  
P3           4     7     2    15    11     4  
  
Average TurnAroundTime=13.750000  
pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6$
```

7. MULTITHREAD SCHEDULING

```

pv@pv-Vostro-5402:/media/pv/Personal/Code/C++/os_lab_observations/lab 6$ ./a.out
<--FCFS Scheduling Algorithm (Non-Preemptive)-->
Enter Number of Processes: 5

Process No: 1
Enter Arrival Time: 4
Enter Burst Time: 7

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

Process No: 3
Enter Arrival Time: 0
Enter Burst Time: 10

Process No: 4
Enter Arrival Time: 6
Enter Burst Time: 13

Process No: 5
Enter Arrival Time: 5
Enter Burst Time: 11

ProcessNo    AT    BT    CT    TAT    WT    RT
P3           0     10    10     10     0     0
P2           3     9     19     16     7     7
P1           4     7     26     22     15    15
P5           5    11     37     32     21    21
P4           6    13     50     44     31    31

Average TurnAroundTime=24.799999
Average WaitingTime=14.800000
<--SJF Scheduling Algorithm (Non-Preemptive)-->
Enter Number of Processes: 5

Process No: 1
Enter Arrival Time: 4
Enter Burst Time: 7

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

Process No: 3
Enter Arrival Time: 0
Enter Burst Time: 10

Process No: 4
Enter Arrival Time: 6
Enter Burst Time: 13

Process No: 5
Enter Arrival Time: 5
Enter Burst Time: 11

Process    AT    BT    CT    TAT    WT    RT
P3         0     10    10     10     0     0
P1         10     7    17     7     0     0
P2         17     9    26     9     0     0
P5         26    11    37    11     0     0
P4         37    13    50    13     0     0

Average TurnAroundTime=10.000000
Average WaitingTime=0.000000
<--Smallest Priority First Scheduling Algorithm (Non-Preemptive)-->
Enter Number of Processes: 5

Process No: 1
Enter Arrival Time: 4
Enter Burst Time: 7
Enter Priority: 3

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

Process No: 3
Enter Arrival Time: 0
Enter Burst Time: 10
Enter Priority: 1

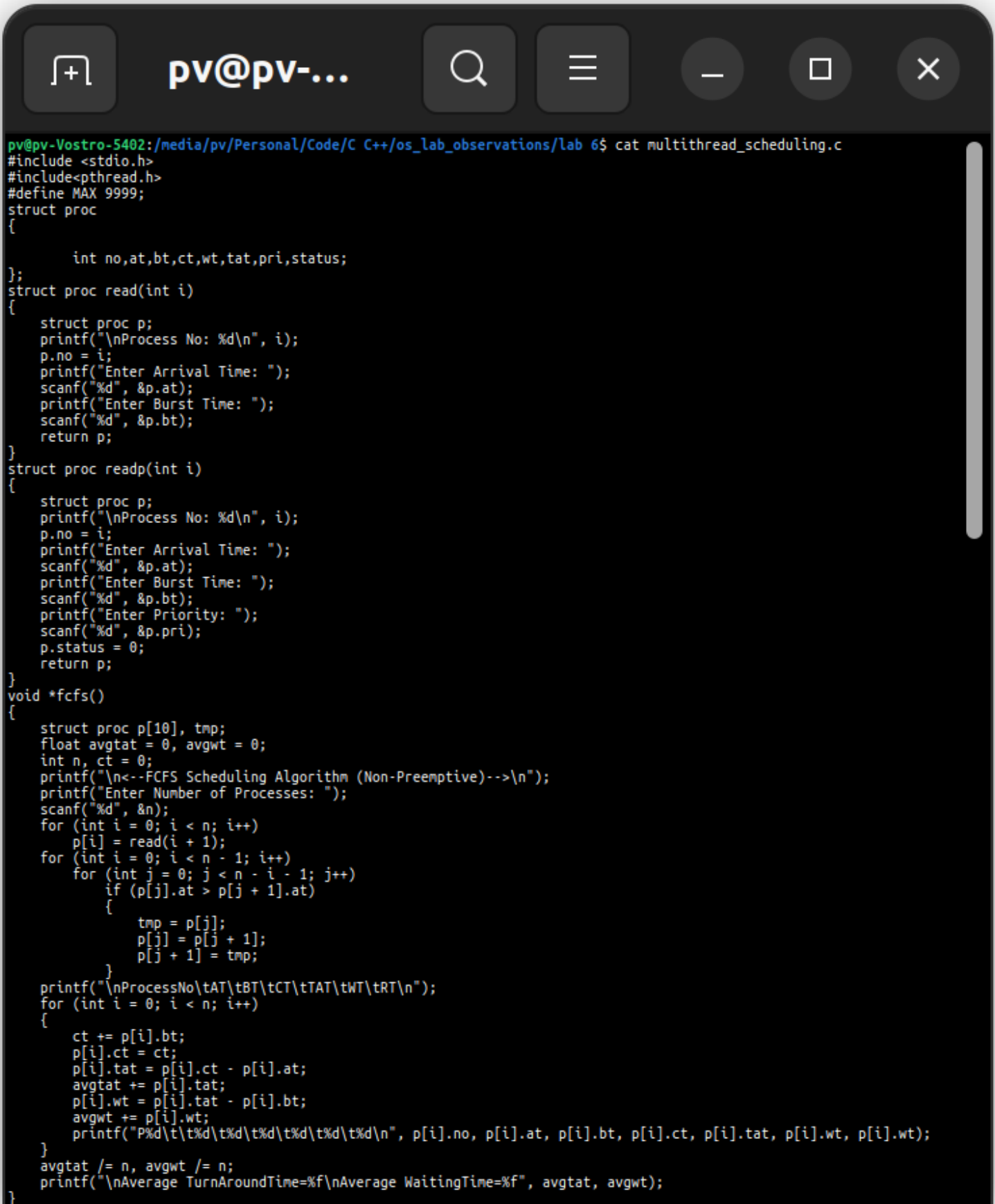
Process No: 4
Enter Arrival Time: 6
Enter Burst Time: 13
Enter Priority: 4

Process No: 5
Enter Arrival Time: 5
Enter Burst Time: 11
Enter Priority: 5

ProcessNo    AT    BT    Pri    CT    TAT    WT    RT
P3           0     10     1     10     10     0     0
P2           3     9      2     19     16     7     7
P1           4     7      3     26     22     15    15
P4           6    13      4     39     33     20    20
P5           5    11      5     50     45     34    34

Average TurnAroundTime=25.200001
pv@pv-Vostro-5402:/media/pv/Personal/Code/C++/os_lab_observations/lab 6$

```



```
pV@pv-...
pv@pv-Vostro-5402:/media/pv/Personal/Code/C++/os_lab_observations/lab 6$ cat multithread_scheduling.c
#include <stdio.h>
#include<pthread.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);
    return p;
}
struct proc readp(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;
}
void *fcfs()
{
    struct proc p[10], tmp;
    float avgtat = 0, avgwt = 0;
    int n, ct = 0;
    printf("\n<--FCFS Scheduling Algorithm (Non-Preemptive)-->\n");
    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)
            {
                tmp = p[j];
                p[j] = p[j + 1];
                p[j + 1] = tmp;
            }
    printf("\nProcessNo\tAT\tBT\tCT\tTAT\tWT\tRT\n");
    for (int i = 0; i < n; i++)
    {
        ct += p[i].bt;
        p[i].ct = ct;
        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%d\t%d\t%d\t%d\t%d\t%d\n", p[i].no, p[i].at, p[i].bt, p[i].ct, p[i].tat, p[i].wt, p[i].wt);
    }
    avgtat /= n, avgwt /= n;
    printf("\nAverage TurnAroundTime=%f\nAverage WaitingTime=%f", avgtat, avgwt);
}
```

```

void *sjf()
{
    int n, j, min = 0;
    float avgtat = 0, avgwt = 0;
    struct proc p[10], temp;
    printf("\n<--SJF Scheduling Algorithm (Non-Preemptive)-->\n");
    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 (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;
            }
    for (j = 1; j < n && p[j].at == p[0].at; j++)
        if (p[j].bt < p[min].bt)
            min = j;
    temp = p[0];
    p[0] = p[min];
    p[min] = temp;
    p[0].at = p[0].at;
    p[0].ct = p[0].at + p[0].bt;
    for (int i = 1; i < n; i++)
    {
        for (j = i + 1, min = i; j < n && p[j].at <= p[i - 1].ct; j++)
            if (p[j].bt < p[min].bt)
                min = j;
        temp = p[i];
        p[i] = p[min];
        p[min] = temp;
        if (p[i].at <= p[i - 1].ct)
            p[i].at = p[i - 1].ct;
        else
            p[i].at = p[i].at;
        p[i].ct = p[i].at + p[i].bt;
    }
    printf("\nProcess\ttAT\tBT\tCT\tTAT\tWT\tRT\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%d\t%d\t%d\t%d\t%d\t%d\n", p[i].no, p[i].at, p[i].bt, p[i].ct, p[i].tat, p[i].wt, p[i].wt);
    }
    avgtat /= n, avgwt /= n;
    printf("\nAverage TurnAroundTime=%f\nAverage WaitingTime=%f", avgtat, avgwt);
}

void *pbs()
{
    int n, s, ct = 0, remaining;
    struct proc p[10], temp;
    float avgtat = 0, avgwt = 0;
    printf("\n<--Smallest Priority First Scheduling Algorithm (Non-Preemptive)-->\n");
    printf("Enter Number of Processes: ");
    scanf("%d", &n);
    for (int i = 0; i < n; i++)
        p[i] = readp(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)
            {

```

```

}
printf("\nProcess\t\tAT\tBT\tCT\tTAT\tWT\tRT\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].ct, p[i].tat, p[i].wt, p[i].wt);
}
avgtat /= n, avgwt /= n;
printf("\nAverage TurnAroundTime=%f\nAverage WaitingTime=%f", avgtat, avgwt);
}
void *pbs()
{
    int n, s, ct = 0, remaining;
    struct proc p[10], temp;
    float avgtat = 0, avgwt = 0;
    printf("\n<--Smallest Priority First Scheduling Algorithm (Non-Preemptive)-->\n");
    printf("Enter Number of Processes: ");
    scanf("%d", &n);
    for (int i = 0; i < n; i++)
        p[i] = readp(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=%f\nAverage WaitingTime=%f", avgtat, avgwt);
}
int main()
{
    pthread_t t1, t2, t3;
    pthread_create(&t1, NULL, (void *)fcfs, NULL);
    pthread_join(t1, NULL);
    pthread_create(&t2, NULL, (void *)sjf, NULL);
    pthread_join(t2, NULL);
    pthread_create(&t3, NULL, (void *)pbs, NULL);
    pthread_join(t3, NULL);

    return 0;
}

```

pv@pv-Vostro-5402:/media/pv/Personal/Code/C C++/os_lab_observations/lab 6\$