LAB:1

1.AIM: write a c program to implement the FCFS program.

PROGRAM:

```
#include<stdio.h>
void main()
{
        int i,j,bt[10],n,wt[10],tt[10],w1=0,t1=0;
        float aw,at;
        printf("enter the 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("\n bt\t wt\t tt\n");
        for(i=0;i<n;i++)
        printf("%d\t%d\t%d\t \n ",bt[i],wt[i],tt[i]);
        printf("aw=%f\n at=%f\n",aw,at);
}
~
```

OUTPUT:

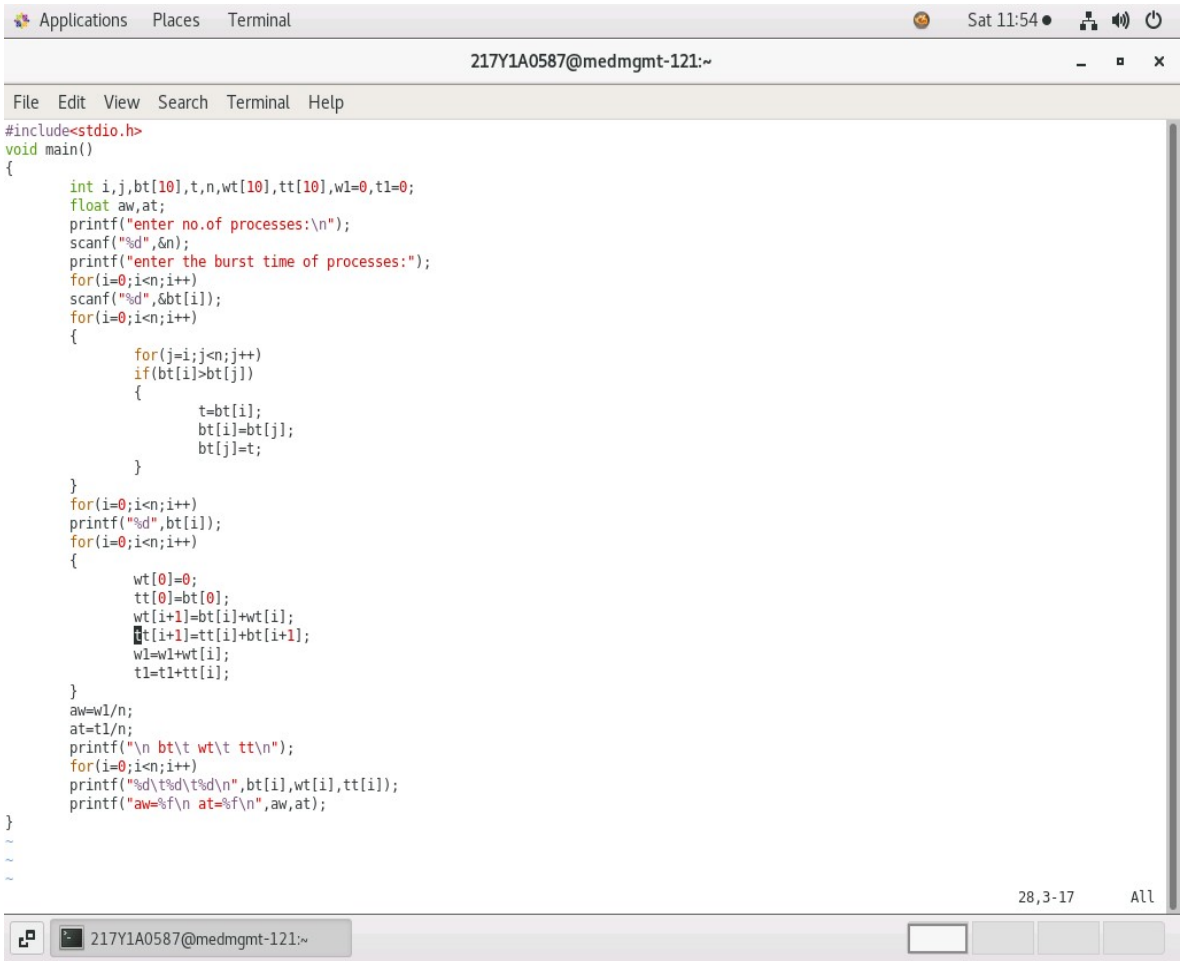
```
[217Y1A0587@medmgmt-121 ~]$ cc fcfs.c
[217Y1A0587@medmgmt-121 ~]$ ./a.out
enter the no.of processes:
3
enter the burst time of processes:
5
6
7

 bt      wt      tt
5       0       5
 6      5       11
 7      11      18
 aw=5.000000
 at=11.000000
[217Y1A0587@medmgmt-121 ~]$
```

LAB:1

2.AIM: write a c program to implement the SJF program.

PROGRAM:



```
Applications  Places  Terminal                                   Sat 11:54
217Y1A0587@medmgmt-121:~
File  Edit  View  Search  Terminal  Help
#include<stdio.h>
void main()
{
        int i,j,bt[10],t,n,wt[10],tt[10],wl=0,tl=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++)
        {
                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];
                wl=wl+wt[i];
                tl=tl+tt[i];
        }
        aw=wl/n;
        at=tl/n;
        printf("\n bt\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);
}
~
~
~
                                                          28,3-17       All
217Y1A0587@medmgmt-121:~
```

OUTPUT:

```
[217Y1A0587@medmgmt-121 ~]$ vi sjf.c
[217Y1A0587@medmgmt-121 ~]$ cc sjf.c
[217Y1A0587@medmgmt-121 ~]$ ./a.out
enter no.of processes:
3
enter the burst time of processes:
6
5
4
456
 bt       wt       tt
4        0        4
5        4        9
6        9        15
aw=4.000000
 at=9.000000
[217Y1A0587@medmgmt-121 ~]$
```

LAB:2

1.AIM: write a c program to implement the Round Robin program.

PROGRAM:

```
#include<stdio.h>
void main()
{
        int st[10],bt[10],wt[10],tat[10],n,tq;
        int i,count=0,swt=0,stat=0,temp,sq=0;
        float awt=0.0,atat=0.0;
        printf("Enter number of processes:");
        scanf("%d",&n);
        printf("Enter burst time for sequences:");
        for(i=0;i<n;i++)
        {
                scanf("%d",&bt[i]);
                st[i]=bt[i];
        }
        printf("Enter time quantum:");
        scanf("%d",&tq);
        while(1)
        {
                for(i=0,count=0;i<n;i++)
                {
                        temp=tq;
                        if(st[i]==0)
                        {
                                count++;
                                continue;
                        }
                        if(st[i]>tq)
                        st[i]=st[i]-tq;
                        else
                        if(st[i]>=0)
                        {
                                temp=st[i];
                                st[i]=0;
                        }
                        sq=sq+temp;
                        tat[i]=sq;
                }
                if(n==count)
                break;
        }
        for(i=0;i<n;i++)
        {
                wt[i]=tat[i]-bt[i];
                swt=swt+wt[i];
                stat=stat+tat[i];
        }
        awt=(float)swt/n;
        atat=(float)stat/n;
        printf("Process_no Burst time Wait time Turn around time");
        for(i=0;i<n;i++)
        printf("\n%d\t %d\t %d\t %d",i+1,bt[i],wt[i],tat[i]);
        printf("\nAvg wait time is %f Avg turn around time is%f",awt,atat);
}
```

OUTPUT:

```
[217Y1A0587@medmgmt-48 ~]$ vi rr.c
[217Y1A0587@medmgmt-48 ~]$ cc rr.c
[217Y1A0587@medmgmt-48 ~]$ ./a.out
Enter number of processes:3
Enter burst time for sequences:
8
10
6
Enter time quantum:2
Process_no Burst time Wait time Turn around time
1        8       12      20
2        10      14      24
3        6       12      18
Avg wait time is 12.666667 Avg turn around time is20.666666
```

LAB:2

2.AIM: write a c program to implement the Priority program.

PROGRAM:

```
#include<stdio.h>
void main()
        {
                int i,j,pno[10],prior[10],bt[10],n,wt[10],tt[10],wl=0,tl=0,s;
                float aw,at;
                printf("enter the number of processes:");
                scanf("%d",&n);
                for(i=0;i<n;i++)
                {
                        printf("The process %d:\n",i+1);
                        printf("Enter the burst time of processes:");
                        scanf("%d",&bt[i]);
                        printf("Enter the priority of processes %d:",i+1);
                        scanf("%d",&prior[i]);
                        pno[i]=i+1;
                }
                for(i=0;i<n;i++)
                {
                        for(j=0;j<n;j++)
                        {
                                if(prior[i]<prior[j])
                                {
                                        s=prior[i];
                                        prior[i]=prior[j];
                                        prior[j]=s;
                                        s=bt[i];
                                        bt[i]=bt[j];
                                        bt[j]=s;
                                        s=pno[i];
                                        pno[i]=pno[j];
                                        pno[j]=s;
                                }
                        }
                }
                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];
                        wl=wl+wt[i];
                        tl=tl+tt[i];
                        aw=wl/n;
                        at=tl/n;
                }
                printf("\n job \t bt \t wt \t tat \t prior\n");
                for(i=0;i<n;i++)
                printf("%d \t %d \t %d\t %d\t %d\n",pno[i],bt[i],wt[i],tt[i],prior[i]);
                printf("aw=%f \t at=%f \n",aw,at);
}
```

OUTPUT:

```
[217Y1A0587@medmgmt-48 ~]$ vi priority.c
[217Y1A0587@medmgmt-48 ~]$ cc priority.c
[217Y1A0587@medmgmt-48 ~]$ ./a.out
enter the number of processes:3
The process 1:
Enter the burst time of processes:4
Enter the priority of processes 1:2
The process 2:
Enter the burst time of processes:7
Enter the priority of processes 2:1
The process 3:
Enter the burst time of processes:8
Enter the priority of processes 3:3

 job      bt       wt       tat      prior
2         7        0        7        1
1         4        7        11       2
3         8        11       19       3
aw=6.000000       at=12.000000
```

LAB:3

1.AIM: write a c program to implement the System calls program.

PROGRAM:

```
#include<unistd.h>
#include<fcntl.h>
#include<string.h>
#include<stdio.h>
int main()
{
        int fd[2];
        char buf1[25]="just for test\n";
        char buf2[100];
        fd[0]=open("tfile.txt",O_RDWR);
        fd[1]=open("tfile.txt",O_RDWR);
        write(fd[0],buf1,strlen(buf1));
        printf("\nenter your test now...");
        scanf("%s",&buf1);
        write(fd[0],buf1,strlen(buf1));
        write(1,buf2,read(fd[1],buf2,sizeof(buf2)));
        close(fd[0]);
        close(fd[1]);
        printf("\n");
        return 0;
}
```

```
just for test
hi
```

OUTPUT:

```
[217Y1A0587@medmgmt-48 ~]$ vi sc.c
[217Y1A0587@medmgmt-48 ~]$ vi tfile.txt
[217Y1A0587@medmgmt-48 ~]$ vi sc.c
[217Y1A0587@medmgmt-48 ~]$ cc sc.c
[217Y1A0587@medmgmt-48 ~]$ ./a.out

enter your test now...hi
just for test
hi
[217Y1A0587@medmgmt-48 ~]$ vi tfile.txt
```