Experiment No. 5

Name : Rinoy Kuriyakose
Roll No: 56

Aim:

Simulate the following CPU scheduling algorithms to find the turn around time and waiting time.

1. FCFS
2. SJF
3. Round Robin (Pre-emptive)
4. Priority.

FCFS Scheduling :

Program:

```
#include<stdio.h>
void main()
 {
   struct process{
      int no;
      float bt;
      float at;
      float tat;
      float wt;
   };
   int n,i,j;
   float time=0,temp1,temp2,temp3,avgtat,avgwt,stat=0,swt=0;
   printf("no. of processes :");
   scanf("%d",&n);
   struct process p[n];
   printf("burst time for the processes :\n");
   for(i=0;i<n;i++){
      printf("p[%d] : ",i+1);
      scanf("%f",&p[i].bt);
   }
   printf("arrival time for the processes :\n");
   for(i=0;i<n;i++){
      printf("p[%d] : ",i+1);
      scanf("%f",&p[i].at);
   }
   for(i=0;i<n;i++){
      p[i].no = i+1;
   }
  for(i=0;i<n;i++){
      for(j=0;j<n-i-1;j++){
        if(p[j].at>p[j+1].at){
           temp1 = p[j].no;
           temp2 = p[j].bt;
           temp3 = p[j].at;
           p[j].no = p[j+1].no;
           p[j].bt = p[j+1].bt;
           p[j].at = p[j+1].at;
```

```
            p[j+1].no = temp1;
            p[j+1].bt = temp2;
            p[j+1].at = temp3;
          }
        }
    }
    for(i=0;i<n;i++){
        if(time<=p[i].at){
            time =p[i].at;
        }
        time = time + p[i].bt;
        p[i].tat = time-p[i].at;
        stat = stat + p[i].tat;
        p[i].wt = p[i].tat - p[i].bt;
        swt = swt + p[i].wt;
    }
    avgtat = stat/n;
    avgwt = swt/n;
      printf("\n  process | burst time  |  arrival time  |  turn-around time  |  waiting time \n");
       printf("  ---------------------------------------------------------------------------------\n");
    for(i=0;i<n;i++){
       printf("  p[%d] : ",p[i].no);
       printf("   %.3f ns",p[i].bt);
       printf("       %.3f ns",p[i].at);
       printf("           %.3f ns",p[i].tat);
       printf("           %.3f ns\n\n",p[i].wt);
    }
    printf(" average turn-around time : %.3f ns\n",avgtat);
    printf(" average waiting time     : %.3f ns\n",avgwt);
 }
```

Output:

```
D:\OS Lab\Day 5 Scheduling>FCFSScheduling
no. of processes :3
burst time for the processes :
p[1] : 8
p[2] : 4
p[3] : 6
arrival time for the processes :
p[1] : 0
p[2] : 0
p[3] : 0

  process | burst time  |  arrival time  |  turn-around time  |  waiting time
  ---------------------------------------------------------------------------
  p[1] :    8.000 ns       0.000 ns           8.000 ns           0.000 ns

  p[2] :    4.000 ns       0.000 ns           12.000 ns           8.000 ns

  p[3] :    6.000 ns       0.000 ns           18.000 ns           12.000 ns

 average turn-around time : 12.667 ns
 average waiting time : 6.667 ns
```

```
D:\OS Lab\Day 5 Scheduling>FCFSScheduling
no. of processes :4
burst time for the processes :
p[1] : 5
p[2] : 6
p[3] : 4
p[4] : 8
arrival time for the processes :
p[1] : 0
p[2] : 0
p[3] : 0
p[4] : 0

  process | burst time   |  arrival time   |  turn-around time   |  waiting time
  ------------------------------------------------------------------------------
  p[1] :    5.000 ns       0.000 ns            5.000 ns            0.000 ns

  p[2] :    6.000 ns       0.000 ns            11.000 ns            5.000 ns

  p[3] :    4.000 ns       0.000 ns            15.000 ns            11.000 ns

  p[4] :    8.000 ns       0.000 ns            23.000 ns            15.000 ns

 average turn-around time : 13.500 ns
 average waiting time : 7.750 ns
```

```
D:\OS Lab\Day 5 Scheduling>FCFSScheduling
no. of processes :4
burst time for the processes :
p[1] : 9
p[2] : 1
p[3] : 8
p[4] : 2
arrival time for the processes :
p[1] : 10
p[2] : 0
p[3] : 2
p[4] : 7

  process | burst time   |  arrival time   |  turn-around time   |  waiting time
  ------------------------------------------------------------------------------
  p[2] :    1.000 ns       0.000 ns            1.000 ns            0.000 ns

  p[3] :    8.000 ns       2.000 ns            8.000 ns            0.000 ns

  p[4] :    2.000 ns       7.000 ns            5.000 ns            3.000 ns

  p[1] :    9.000 ns       10.000 ns            11.000 ns            2.000 ns

 average turn-around time : 6.250 ns
 average waiting time : 1.250 ns
```

SJF Scheduling :

Program:

```
include<stdio.h>
void main()
 {
   struct process{
      int no;
      float bt;
      float at;
      float tat;
      float wt;
   };
   int n,i,j,d,k=1;
   float time=0,temp1,temp2,temp3,avgtat,avgwt,stat=0,swt=0,min,sum,btime;
   printf("no. of processes :");
   scanf("%d",&n);
   struct process p[n];
   printf("burst time for the processes :\n");
   for(i=0;i<n;i++){
      printf("p[%d] : ",i+1);
      scanf("%f",&p[i].bt);
   }
   printf("arrival time for the processes :\n");
   for(i=0;i<n;i++){
      printf("p[%d] : ",i+1);
      scanf("%f",&p[i].at);
   }
   for(i=0;i<n;i++){
      p[i].no = i+1;
   }
   for(i=0;i<n;i++){
      for(j=0;j<n-i-1;j++){
        if(p[j].bt>p[j+1].bt){
           temp1 = p[j].no;
           temp2 = p[j].bt;
           temp3 = p[j].at;
           p[j].no = p[j+1].no;
           p[j].bt = p[j+1].bt;
           p[j].at = p[j+1].at;
           p[j+1].no = temp1;
           p[j+1].bt = temp2;
           p[j+1].at = temp3;
        }
      }
   }
   for(i=0;i<n;i++){
      for(j=0;j<n-i-1;j++){
        if(p[j].at>p[j+1].at){
           temp1 = p[j].no;
           temp2 = p[j].bt;
```

```
            temp3 = p[j].at;
            p[j].no = p[j+1].no;
            p[j].bt = p[j+1].bt;
            p[j].at = p[j+1].at;
            p[j+1].no = temp1;
            p[j+1].bt = temp2;
            p[j+1].at = temp3;
          }
       }
    }
    for(j=0;j<n;j++){
       btime=btime+p[j].bt;
       min=p[k].bt;
       for(i=k;i<n;i++){
          if ((btime>=p[i].at) && (p[i].bt<min)){
              min = p[i].bt;
              temp1 = p[i].no;
              temp2 = p[i].bt;
              temp3 = p[i].at;
              p[i].no = p[k].no;
              p[i].bt = p[k].bt;
              p[i].at = p[k].at;
              p[k].no = temp1;
              p[k].bt = temp2;
              p[k].at = temp3;
           }
       }
       k++;
    }
    sum = p[0].at;
    for(i=0;i<n;i++){
       if(sum<p[i].at){
          sum=p[i].at;
       }
       sum = sum + p[i].bt;
    p[i].tat=sum-p[i].at;
    p[i].wt=p[i].tat-p[i].bt;
    stat=stat+p[i].tat;
    swt=swt+p[i].wt;
    }
    avgtat = stat/n;
    avgwt = swt/n;
    printf("\n  process | burst time  |  arrival time  |  turn-around time  |  waiting time \n");
       printf("  ---------------------------------------------------------------------------\n");
    for(i=0;i<n;i++){
       printf("  p[%d] : ",p[i].no);
       printf("   %.3f ns",p[i].bt);
       printf("       %.3f ns",p[i].at);
       printf("            %.3f ns",p[i].tat);
       printf("            %.3f ns\n\n",p[i].wt);
    }
    printf(" average turn-around time : %.3f ns\n",avgtat);
```

```
    printf(" average waiting time     : %.3f ns\n",avgwt);
}
```

Output:

```
D:\OS Lab\Day 5 Scheduling>SJFScheduling
no. of processes :4
burst time for the processes :
p[1] : 6
p[2] : 1
p[3] : 5
p[4] : 2
arrival time for the processes :
p[1] : 0
p[2] : 1
p[3] : 1
p[4] : 2

  process | burst time  |  arrival time  |  turn-around time  |  waiting time
  ------------------------------------------------------------------------------
  p[1] :    6.000 ns       0.000 ns           6.000 ns           0.000 ns

  p[2] :    1.000 ns       1.000 ns           6.000 ns           5.000 ns

  p[4] :    2.000 ns       2.000 ns           7.000 ns           5.000 ns

  p[3] :    5.000 ns       1.000 ns           13.000 ns           8.000 ns

 average turn-around time : 8.000 ns
 average waiting time     : 4.500 ns
```

```
D:\OS Lab\Day 5 Scheduling>SJFScheduling
no. of processes :3
burst time for the processes :
p[1] : 6
p[2] : 4
p[3] : 8
arrival time for the processes :
p[1] : 0
p[2] : 0
p[3] : 0

  process | burst time  |  arrival time  |  turn-around time  |  waiting time
  ------------------------------------------------------------------------------
  p[2] :    4.000 ns       0.000 ns           4.000 ns           0.000 ns

  p[1] :    6.000 ns       0.000 ns           10.000 ns           4.000 ns

  p[3] :    8.000 ns       0.000 ns           18.000 ns           10.000 ns

 average turn-around time : 10.667 ns
 average waiting time     : 4.667 ns
```

```
D:\OS Lab\Day 5 Scheduling>SJFScheduling
no. of processes :3
burst time for the processes :
p[1] : 10
p[2] : 3
p[3] : 1
arrival time for the processes :
p[1] : 0
p[2] : 1
p[3] : 2

  process | burst time   |  arrival time   |  turn-around time   |  waiting time
  ---------------------------------------------------------------------------
  p[1] :    10.000 ns       0.000 ns            10.000 ns            0.000 ns

  p[3] :    1.000 ns       2.000 ns            9.000 ns            8.000 ns

  p[2] :    3.000 ns       1.000 ns            13.000 ns            10.000 ns

 average turn-around time : 10.667 ns
 average waiting time     : 6.000 ns
```

Round Robin Scheduling :

Program:

```
#include<stdio.h>
void main()
 {
   struct process{
      int no;
      float bt;
      float at;
      float tat;
      float wt;
   };
   int n,i,j,quantum,count,var;
   float time=0,temp1,temp2,temp3,temp4,avgtat,avgwt,stat=0,swt=0,sum;
   printf(" no. of processes :");
   scanf("%d",&n);
   struct process p[n];
   float temp[n];
   printf(" burst time for the processes :\n");
   for(i=0;i<n;i++){
      printf(" p[%d] : ",i+1);
      scanf("%f",&p[i].bt);
      temp[i]=p[i].bt;
   }
   printf(" arrival time for the processes :\n");
   for(i=0;i<n;i++){
      printf(" p[%d] : ",i+1);
      scanf("%f",&p[i].at);
   }
```

```
 for(i=0;i<n;i++){
    p[i].no = i+1;
 }
 printf(" time quantum for the processes: ");
 scanf("%d", &quantum);
for(i=0;i<n;i++){
    for(j=0;j<n-i-1;j++){
      if(p[j].at>p[j+1].at){
         temp1 = p[j].no;
         temp2 = p[j].bt;
         temp3 = p[j].at;
         temp4 = temp[j];
         p[j].no = p[j+1].no;
         p[j].bt = p[j+1].bt;
         p[j].at = p[j+1].at;
         temp[j] = temp[j+1];
         p[j+1].no = temp1;
         p[j+1].bt = temp2;
         p[j+1].at = temp3;
         temp[j+1] = temp4;
      }
    }
 }
 sum=p[0].at;
 var=n;
 for( i = 0; var!=0; ){
    if(temp[i] <= quantum && temp[i] > 0){
       sum = sum + temp[i];
       temp[i] = 0;
       count=1;
    }
    else if(temp[i] > 0){
       temp[i] = temp[i] - quantum;
       sum = sum + quantum;
    }
    if(temp[i]==0 && count==1){
       var--;
       p[i].tat=sum-p[i].at;
       p[i].wt=sum-p[i].at-p[i].bt;
       swt = swt+p[i].wt;
       stat = stat+p[i].tat;
       count =0;
    }
    if(i==n-1){
       i=0;
    }
    else if(p[i+1].at<=sum){
       i++;
    }
    /*else if(p[i+1].at>=sum){
       sum=p[i+1].at;
       i++;
```

```
    }*/
    else{
       i=0;
    }
  }
  avgtat = stat/n;
  avgwt = swt/n;
    printf("\n  process | burst time  |  arrival time  |  turn-around time  |  waiting time \n");
     printf("  ---------------------------------------------------------------------------\n");
  for(i=0;i<n;i++){
     printf("  p[%d] : ",p[i].no);
     printf("   %.3f ms",p[i].bt);
     printf("       %.3f ms",p[i].at);
     printf("           %.3f ms",p[i].tat);
     printf("           %.3f ms\n\n",p[i].wt);
  }
  printf(" average turn-around time : %.3f ms\n",avgtat);
  printf(" average waiting time     : %.3f ms\n",avgwt);
}
```

Output:

```
D:\OS Lab\Day 5 Scheduling>RRScheduling
 no. of processes :4
 burst time for the processes :
 p[1] : 5
 p[2] : 6
 p[3] : 3
 p[4] : 8
 arrival time for the processes :
 p[1] : 0
 p[2] : 0
 p[3] : 0
 p[4] : 0
 time quantum for the processes: 2

  process | burst time  |  arrival time  |  turn-around time  |  waiting time
  ---------------------------------------------------------------------------
  p[1] :    5.000 ms       0.000 ms           16.000 ms           11.000 ms

  p[2] :    6.000 ms       0.000 ms           18.000 ms           12.000 ms

  p[3] :    3.000 ms       0.000 ms           13.000 ms           10.000 ms

  p[4] :    8.000 ms       0.000 ms           22.000 ms           14.000 ms

 average turn-around time : 17.250 ms
 average waiting time     : 11.750 ms
```

```
D:\OS Lab\Day 5 Scheduling>RRScheduling
 no. of processes :3
 burst time for the processes :
 p[1] : 8
 p[2] : 4
 p[3] : 6
 arrival time for the processes :
 p[1] : 0
 p[2] : 0
 p[3] : 0
 time quantum for the processes: 4

  process | burst time   |  arrival time  |  turn-around time  |  waiting time
  ---------------------------------------------------------------------------
  p[1] :    8.000 ms       0.000 ms           16.000 ms            8.000 ms

  p[2] :    4.000 ms       0.000 ms           8.000 ms           4.000 ms

  p[3] :    6.000 ms       0.000 ms           18.000 ms            12.000 ms

 average turn-around time : 14.000 ms
 average waiting time     : 8.000 ms
```

```
D:\OS Lab\Day 5 Scheduling>RRScheduling
 no. of processes :4
 burst time for the processes :
 p[1] : 13
 p[2] : 9
 p[3] : 5
 p[4] : 7
 arrival time for the processes :
 p[1] : 0
 p[2] : 2
 p[3] : 2
 p[4] : 3
 time quantum for the processes: 3

  process | burst time   |  arrival time  |  turn-around time  |  waiting time
  ---------------------------------------------------------------------------
  p[1] :    13.000 ms       0.000 ms           34.000 ms            21.000 ms

  p[2] :    9.000 ms       2.000 ms           27.000 ms           18.000 ms

  p[3] :    5.000 ms       2.000 ms           18.000 ms           13.000 ms

  p[4] :    7.000 ms       3.000 ms           27.000 ms           20.000 ms

 average turn-around time : 26.500 ms
 average waiting time     : 18.000 ms
```

Priority Scheduling :

Program:

```
#include<stdio.h>
void main()
 {
   struct process{
      int no;
      float bt;
      float at;
      float tat;
      float wt;
      int pr;
   };
   int n,i,j,d,k=1,min,temp4;
   float time=0,temp1,temp2,temp3,avgtat,avgwt,stat=0,swt=0,sum,btime;
   printf("no. of processes :");
   scanf("%d",&n);
   struct process p[n];
   printf("burst time for the processes :\n");
   for(i=0;i<n;i++){
      printf("p[%d] : ",i+1);
      scanf("%f",&p[i].bt);
   }
   printf("arrival time for the processes :\n");
   for(i=0;i<n;i++){
      printf("p[%d] : ",i+1);
      scanf("%f",&p[i].at);
   }
   printf("priority for the processes :\n");
   for(i=0;i<n;i++){
      printf("p[%d] : ",i+1);
      scanf("%d",&p[i].pr);
   }
   for(i=0;i<n;i++){
      p[i].no = i+1;
   }
   for(i=0;i<n;i++){
      for(j=0;j<n-i-1;j++){
         if(p[j].pr>p[j+1].pr){
            temp1 = p[j].no;
            temp2 = p[j].bt;
            temp3 = p[j].at;
            temp4 = p[j].pr;
            p[j].no = p[j+1].no;
            p[j].bt = p[j+1].bt;
            p[j].at = p[j+1].at;
            p[j].pr = p[j+1].pr;
            p[j+1].no = temp1;
            p[j+1].bt = temp2;
            p[j+1].at = temp3;
```

```
            p[j+1].pr = temp4;
        }
    }
}
for(i=0;i<n;i++){
    for(j=0;j<n-i-1;j++){
        if(p[j].at>p[j+1].at){
            temp1 = p[j].no;
            temp2 = p[j].bt;
            temp3 = p[j].at;
            temp4 = p[j].pr;
            p[j].no = p[j+1].no;
            p[j].bt = p[j+1].bt;
            p[j].at = p[j+1].at;
            p[j].pr = p[j+1].pr;
            p[j+1].no = temp1;
            p[j+1].bt = temp2;
            p[j+1].at = temp3;
            p[j+1].pr = temp4;
        }
    }
}
for(j=0;j<n;j++){
    btime=btime+p[j].bt;
    min=p[k].pr;
    for(i=k;i<n;i++){
        if ((btime>=p[i].at) && (p[i].pr<min)){
            min = p[i].pr;
            temp1 = p[i].no;
            temp2 = p[i].bt;
            temp3 = p[i].at;
            temp4 = p[i].pr;
            p[i].no = p[k].no;
            p[i].bt = p[k].bt;
            p[i].at = p[k].at;
            p[i].pr = p[k].pr;
            p[k].no = temp1;
            p[k].bt = temp2;
            p[k].at = temp3;
            p[k].pr = temp4;
        }
    }
    k++;
}
sum = p[0].at;
for(i=0;i<n;i++){
    if(sum<p[i].at){
        sum=p[i].at;
    }
    sum = sum + p[i].bt;
p[i].tat=sum-p[i].at;
p[i].wt=p[i].tat-p[i].bt;
```

```
    stat=stat+p[i].tat;
    swt=swt+p[i].wt;
    }
    avgtat = stat/n;
    avgwt = swt/n;
    printf("\n process | burst time  |  arrival time  |   priority   | turn-around time  |  waiting time \n");
    printf(" ----------------------------------------------------------------------------------------\n");
    for(i=0;i<n;i++){
        printf("  p[%d] : ",p[i].no);
        printf("   %.3f ns",p[i].bt);
        printf("       %.3f ns",p[i].at);
         printf("          %d",p[i].pr);
        printf("             %.3f ns",p[i].tat);
        printf("             %.3f ns\n\n\n",p[i].wt);
    }
    printf(" average turn-around time : %.3f ns\n",avgtat);
    printf(" average waiting time     : %.3f ns\n",avgwt);
}
```

Output:

```
D:\OS Lab\Day 5 Scheduling>PriorityScheduling
no. of processes :4
burst time for the processes :
p[1] : 5
p[2] : 4
p[3] : 1
p[4] : 2
arrival time for the processes :
p[1] : 0
p[2] : 2
p[3] : 3
p[4] : 5
priority for the processes :
p[1] : 3
p[2] : 1
p[3] : 2
p[4] : 4

 process | burst time  |  arrival time  |   priority   | turn-around time  |  waiting time
 ----------------------------------------------------------------------------------------
  p[1] :    5.000 ns       0.000 ns          3              5.000 ns             0.000 ns

  p[2] :    4.000 ns       2.000 ns          1              7.000 ns             3.000 ns

  p[3] :    1.000 ns       3.000 ns          2              7.000 ns             6.000 ns

  p[4] :    2.000 ns       5.000 ns          4              7.000 ns             5.000 ns

 average turn-around time : 6.500 ns
 average waiting time     : 3.500 ns
```

```
D:\OS Lab\Day 5 Scheduling>PriorityScheduling
no. of processes :4
burst time for the processes :
p[1] : 5
p[2] : 6
p[3] : 3
p[4] : 8
arrival time for the processes :
p[1] : 0
p[2] : 0
p[3] : 0
p[4] : 0
priority for the processes :
p[1] : 3
p[2] : 1
p[3] : 4
p[4] : 2

 process | burst time   |  arrival time  |   priority   |  turn-around time  |  waiting time
 ------------------------------------------------------------------------------------------
  p[2] :    6.000 ns       0.000 ns          1              6.000 ns             0.000 ns

  p[4] :    8.000 ns       0.000 ns          2              14.000 ns             6.000 ns

  p[1] :    5.000 ns       0.000 ns          3              19.000 ns             14.000 ns

  p[3] :    3.000 ns       0.000 ns          4              22.000 ns             19.000 ns

 average turn-around time : 15.250 ns
 average waiting time     : 9.750 ns
```

```
D:\OS Lab\Day 5 Scheduling>PriorityScheduling
no. of processes :5
burst time for the processes :
p[1] : 11
p[2] : 28
p[3] : 2
p[4] : 10
p[5] : 16
arrival time for the processes :
p[1] : 0
p[2] : 5
p[3] : 12
p[4] : 2
p[5] : 9
priority for the processes :
p[1] : 2
p[2] : 0
p[3] : 3
p[4] : 1
p[5] : 4

 process | burst time   |  arrival time  |   priority   |  turn-around time  |  waiting time
 ------------------------------------------------------------------------------------------
  p[1] :    11.000 ns       0.000 ns          2              11.000 ns             0.000 ns

  p[2] :    28.000 ns       5.000 ns          0              34.000 ns             6.000 ns

  p[4] :    10.000 ns       2.000 ns          1              47.000 ns             37.000 ns

  p[3] :    2.000 ns       12.000 ns          3              39.000 ns             37.000 ns

  p[5] :    16.000 ns       9.000 ns          4              58.000 ns             42.000 ns

 average turn-around time : 37.800 ns
 average waiting time     : 24.400 ns
```