

IMPLEMENTATION OF CPU SCHEDULING ALGORITHMS

A. FIRST COME, FIRST SERVE SCHEDULING

```
#include<stdio.h>
struct proc
{
    int a;
    int b;
    int no;
    int wt ;
    int tat ;
};
struct proc read(int i)
{
    struct proc p;
    printf("\n\n The process no.:%d.\n",i);
    p.no=i;
    printf("Enter the arrival time:");
    scanf("%d",&p.a);
    printf("Enter the burst time:");
    scanf("%d",&p.b);
    return p;
}

int main()
{
    struct proc p[10],temp;
    int n, g[10];
    float att=0,awt=0;
    printf("enter the number of process");
    scanf("%d",&n);

    int i,j;
    for(i=0;i<n;i++)
    p[i]=read(i);
    for(i=0;i<n;i++)
    {
        for(j=i;j<n;j++)
        {
            if(p[i].a>p[j].a)
            {
```

```

        temp=p[i];
        p[i]=p[j];
        p[j]=temp;
    }
}

```

```

g[0]=0;
for(i=0;i<n;i++)
    g[i+1]=g[i]+p[i].b;
for(i=0;i<n;i++)
{
    p[i].wt=g[i]-p[i].a;
    p[i].tat=g[i+1]-p[i].a;
    awt=awt+p[i].wt;
    att=att+p[i].tat;
}
awt =awt/n;
att=att/n;
printf("\n\tprocess\tarrival time \t burst time\twaiting time\tturn around
time\n");
for(i=0;i<n;i++)
{
    printf("\tp%d\t\t%d\t\t%d\t\t%d\t\t%d\n",i,p[i].a,p[i].b,p[i].wt,p[i].tat);
}
printf("the average waiting time is %f\n",awt);
printf("the average turn around time is %f\n",att);
return 0;
}

```

OUTPUT:

enter the number of process3

The process no.:0.

Enter the arrival time:1

Enter the burst time:25

The process no.:1.

Enter the arrival time:5

Enter the burst time:5

The process no.:2.
Enter the arrival time:0
Enter the burst time:10

process	arrival time	burst time	waiting time	turn around time
p0	0	10	0	10
p1	1	25	9	34
p2	5	5	30	35

the average waiting time is 13.000000
the average turn around time is 26.333334

B.SHORTEST JOB FIRST SCHEDULING

PROGRAM:

```
#include<stdio.h>
int main()
{
    int n,j,temp,temp1,temp2,pr[10],b[10],t[10],w[10],p[10],i,ar[10],s[10],g[10];

    float att=0,awt=0;

    for(i=0;i<10;i++)
    {
        b[i]=0;w[i]=0;
    }

    printf("enter the number of process:");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        p[i]=i;
        printf("enter the burst time of p[%d]:",p[i]);
        scanf("%d",&b[i]);
        printf("enter the arrival time of p[%d]:",p[i]);scanf("%d",&ar[i]);
    }
    for(i=0;i<n;i++)
    {
        for(j=i;j<n;j++)
        {
            if(ar[i]>ar[j])
            {
                temp2=ar[i];
```

```

        ar[i]=ar[j];
        ar[j]=temp2;
        temp1=p[i];
        p[i]=p[j];
        p[j]=temp1;temp=b[i]; b[i]=b[j]; b[j]=temp;
    }
    else if(ar[i]==ar[j])
    {
        if(b[i]>b[j])
        {
            temp2=ar[i];
            ar[i]=ar[j];
            ar[j]=temp2;
            temp1=p[i];
            p[i]=p[j];
            p[j]=temp1;temp=b[i]; b[i]=b[j]; b[j]=temp;
        }
    }
    else
    {
    }
}

printf("\nprocess name\tbursttime\tarrivaltime\n");
for(i=0;i<n;i++)
printf("\n p[%d]\t\t %d \t\t %d\n",p[i],b[i],ar[i]);
w[0]=0;
for(i=0;i<n;i++)
w[i+1]=w[i]+b[i];
for(i=0;i<n;i++)
{
    s[i]=w[i]+b[i];
    t[i]=(w[i]+b[i]-ar[i]);
    awt=awt+w[i];
    att=att+t[i];
}
awt=awt/n;
att=att/n;
printf("\n\t process \t waiting time \t turn around time \n");
for(i=0;i<n;i++)
printf("\t p[%d] \t\t\t %d \t\t\t %d \n",p[i],w[i],t[i]);
printf("\n GANNT CHART:");

```

```

for(i=0;i<n;i++)
{
    printf("\tp[%d]\t",p[i]);
}
printf("\n");

for(i=0;i<n;i++)
{
    g[i]=s[i];
}
printf("0");
for(i=0;i<n;i++)
{
    printf("\t\t%d",g[i]);
}
printf("\nthe average waitingtimeis %f:\n",awt);
printf("the average turn around time is %f:\n",att);
return 1;
}

```

OUTPUT:

```

enter the number of process:3
enter the burst time of p[0]:2
enter the arrival time of p[0]:0
enter the burst time of p[1]:5
enter the arrival time of p[1]:0
enter the burst time of p[2]:3
enter the arrival time of p[2]:0

```

processname	bursttime	arrivaltime
p[0]	2	0
p[2]	3	0
p[1]	5	0

process	waiting time	turn around time
p[0]	0	2

p[2]	2	5
p[1]	5	10

GANTT CHART:

	p[0]		p[2]		p[1]	
0		2		5		10

the average waiting time is 2.333333

the average turn around time is 5.666667

C.PRIORITY SCHEDULING

PROGRAM:

```
#include<stdio.h>
```

```
int main()
{
```

```
int n,j,temp,temp1,temp2,pr[10],b[10],t[10],w[10],p[10],i,ar[10],s[10],g[10];
```

```
float att=0,awt=0;
```

```
for(i=0;i<10;i++)
```

```
{
```

```
    b[i]=0;w[i]=0;
```

```
}
```

```
printf("enter the number of process:");
```

```
scanf("%d",&n);
```

```
for(i=0;i<n;i++)
```

```
{
```

```
    p[i]=i;
```

```
    printf("enter the burst time of p[%d]:",p[i]);
```

```
    scanf("%d",&b[i]);
```

```
    printf("enter the priority value of p[%d]:",p[i]);scanf("%d",&ar[i]);
```

```
}
```

```
for(i=0;i<n;i++)
```

```
{
```

```
    for(j=i;j<n;j++)
```

```
{
```

```
        if(ar[i]>ar[j])
```

```
{
```

```

        temp2=ar[i];
        ar[i]=ar[j];
        ar[j]=temp2;
        temp1=p[i];
        p[i]=p[j];
        p[j]=temp1;temp=b[i]; b[i]=b[j]; b[j]=temp;
    }
    else if(ar[i]==ar[j])
    {
        if(b[i]>b[j])
        {
            temp2=ar[i];
            ar[i]=ar[j];
            ar[j]=temp2;
            temp1=p[i];
            p[i]=p[j];
            p[j]=temp1;temp=b[i]; b[i]=b[j]; b[j]=temp;
        }
    }
    else
    {
    }
}

printf("\nprocess name\tbursttime\tpriority\n");
for(i=0;i<n;i++)
printf("\n p[%d]\t\t %d \t\t %d\n",p[i],b[i],ar[i]);
w[0]=0;
for(i=0;i<n;i++)
w[i+1]=w[i]+b[i];
for(i=0;i<n;i++)
{
    s[i]=w[i]+b[i];
    t[i]=(w[i]+b[i]);
    awt=awt+w[i];
    att=att+t[i];
}
awt=awt/n;
att=att/n;
printf("\n\t process \t waiting time \t turn around time \n");
for(i=0;i<n;i++)
printf("\t p[%d] \t\t\t %d \t\t\t %d \n",p[i],w[i],t[i]);

```

```

printf("\n GANNT CHART:");
for(i=0;i<n;i++)
{
    printf("\tp[%d]\t",p[i]);
}
printf("\n");

for(i=0;i<n;i++)
{
    g[i]=s[i];
}
printf("0");
for(i=0;i<n;i++)
{
    printf("\t\t%d",g[i]);
}
printf("\nthe average waitingtimeis %f:\n",awt);
printf("the average turn around time is %f\n:",att);
return 1;
}

```

OUTPUT:

```

enter the number of process:3
enter the burst time of p[0]:1
enter the priority value of p[0]:2
enter the burst time of p[1]:3
enter the priority value of p[1]:3
enter the burst time of p[2]:6
enter the priority value of p[2]:0

```

process name	bursttime	priority
p[2]	6	0
p[0]	1	2
p[1]	3	3

process	waiting time	turn around time
p[2]	0	6

p[0]	6	7
p[1]	7	10

GANTT CHART:

0	p[2]	6	p[0]	7	p[1]	10
---	------	---	------	---	------	----

the average waiting time is 4.333333

the average turn around time is 7.666667

D.ROUND ROBIN SCHEDULING

PROGRAM:

```
#include<stdio.h>
struct process
{
    char pname[10];
    int bt,rt,wt,tt;
} p[10],t;
void main()
{
    int t[15];
    int temp,tq,n,i;
    int count,sq,times=0;
    float sum=0.0,sum1=0.0;
    printf("Enter the value for quantum time:");
    scanf("%d",&tq);
    printf("Enter the value of n:");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter the name:");
        scanf("%s",&p[i].pname);
        printf("Enter the burst time:");
        scanf("%d",&p[i].bt);
    }
    for(i=0;i<n;i++)
    {
        p[i].rt=p[i].bt;
    }
    printf("Gantt chart:\n");
```

```

        sq=0;
        while(1)
        {
for(i=0,count=0;i<n;i++)
        {

                temp=tq;
                if(p[i].rt==0)
                {
                        count++;
                        continue;
                }
                else
                {
                        if(p[i].rt>=tq)
                        {
                                p[i].rt=p[i].rt-tq;
                        }
                        else if(p[i].rt>0)
                        {
                                temp=p[i].rt;
                                p[i].rt=0;
                        }
                        t[times++]=sq;
                        sq=sq+temp;
                        p[i].tt=sq;
                }
                printf("\t %s",p[i].pname);
                t[times]=sq;

        }
        if(count==n)
        break;
}

```

```

for(i=0;i<n;i++)
{
        p[i].wt=p[i].tt-p[i].bt;
        sum=sum+p[i].wt;
        sum1=sum1+p[i].tt;
}

```

```

printf("\n");
for(i=0;i<=times;i++)
{
    printf("\t%d",t[i]);

}

printf("\n The average waiting time=%f",sum/n);
printf("\n The average turn around time=%f",sum1/n);

}

```

OUTPUT:

Enter the value for quantum time2
Enter the value of n3
Enter the namep1
Enter the burst time3
Enter the namep2
Enter the burst time5
Enter the namep3
Enter the burst time2

Gantt chart

p1	p2	p3	p1	p2	p2	
0	2	4	6	7	9	10

The average waiting time=4.333333
The average turn around time=7.666667