

Practical 4

Name: Rahul Baser

Roll No.: A75

Aim: SJF and Priority Scheduling

SJF code

```
#include<stdio.h>

struct time
{
    int p,at,bt,wt,tat,st;
};

int process(struct time a[], int n,int t)
{
    int i,m, mintime=999;
    for(i=0;i<n;i++)
    {
        if(a[i].at <= t && a[i].st == 0)
        {
            if(mintime > a[i].bt)
            {
                mintime = a[i].bt;
                m = i;
            }
        }
    }
    a[m].st = 1;
    return m;
}

void ganttchat(struct time a[],int gc[],int m)
{
    int i,x=0;
    printf("Gantt Chat\n\n");
    printf("|\\t");
    for(i=0;i<m;i++)
    {
        printf("P%d\\t\\t", a[gc[i]].p);
    }
    printf("\\n");
    printf("0\\t");
}
```

```

        for(i=0;i<m;i++)
        {
            x = x + a[gc[i]].bt;
            printf("\t%d\t", x);
        }
        printf("\n");
        return;
    }

int main()
{
    int i,n = 5,cp,t=0,gc[100];
    struct time a[5];
    float avgwt=0,avgtt=0;

    a[0].p = 0;
    a[1].p = 1;
    a[2].p = 2;
    a[3].p = 3;
    a[4].p = 4;

    a[0].st = 0;
    a[1].st = 0;
    a[2].st = 0;
    a[3].st = 0;
    a[4].st = 0;

    a[0].at = 2;
    a[1].at = 5;
    a[2].at = 1;
    a[3].at = 0;
    a[4].at = 4;

    a[0].bt = 6;
    a[1].bt = 2;
    a[2].bt = 8;
    a[3].bt = 3;
    a[4].bt = 4;

    for(i=0;i<n;i++)
    {
        cp = process(a,n,t);
        a[cp].wt = t - a[cp].at;
        a[cp].tat = a[cp].at + a[cp].bt;
        t = t + a[cp].bt;
        avgwt = avgwt + a[cp].wt;
        avgtt = avgtt + a[cp].tat;
    }
}

```


Preemptive priority code

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

```
struct proc{  
    int a_t;  
    int b_t;  
    int w_t;  
    int priority;  
};
```

```
int main(){
```

```
    struct proc s[7];  
    int a[7];  
    int count=0;  
    int cur=0;  
    int c=0;  
    int time=0;  
    int total=0;  
    int t=7;
```

```
    s[0].a_t=0;  
    s[1].a_t=2;  
    s[2].a_t=1;  
    s[3].a_t=4;  
    s[4].a_t=6;  
    s[5].a_t=5;  
    s[6].a_t=7;
```

```
    s[0].b_t=3;  
    s[1].b_t=5;  
    s[2].b_t=4;  
    s[3].b_t=2;  
    s[4].b_t=9;  
    s[5].b_t=4;  
    s[6].b_t=10;
```

```
    s[0].priority=2;  
    s[1].priority=6;  
    s[2].priority=3;  
    s[3].priority=5;  
    s[4].priority=7;
```

```

s[5].priority=4;
s[6].priority=10;

printf(" Process | Arrival Time | Burst Time | Priority |\n");
for(int i=0;i<7;i++)
{
printf("  P%d |      %d      |      %d      |      %d
|\n",i+1,s[i].a_t,s[i].b_t,s[i].priority);
}

int i=0;
while(i<7)
{
    total+=s[i].b_t;
    i++;
}
printf("\n");

while(cur<=total)
{
    count=0;
    for(i=0;i<7;i++)
    {
        if(s[i].a_t<=cur)
        {
            if(s[i].b_t>0)
            {
                a[count]=i;
                count++;
            }
        }
    }

    i=0;
    int min=1000;
    while(i<count)
    {
        if(min>s[a[i]].priority)
        {
            min=s[a[i]].priority;
            c=a[i];
        }
        i++;
    }

    printf("  %d |  P%d  |",cur,c+1);
    s[c].b_t--;

```

```

        if(s[c].b_t==1)
        {
            s[c].a_t=1000;
        }
        cur++;
    }

    printf("  %d  |",cur);
    return 0;
}

```

Output

```

Process | Arrival Time | Burst Time | Priority |
P1 | 0 | 3 | 2 |
P2 | 2 | 5 | 6 |
P3 | 1 | 4 | 3 |
P4 | 4 | 2 | 5 |
P5 | 6 | 9 | 7 |
P6 | 5 | 4 | 4 |
P7 | 7 | 10 | 10 |

0 | P1 | 1 | P1 | 2 | P3 | 3 | P3 | 4 | P3 | 5 | P6 | 6 | P6 | 7 | P6 | 8 | P4 | 9 | P2 |
| 10 | P2 | 11 | P2 | 12 | P2 | 13 | P5 | 14 | P5 | 15 | P5 | 16 | P5 | 17 | P5 | 18 | P5 |
19 | P5 | 20 | P5 | 21 | P7 | 22 | P7 | 23 | P7 | 24 | P7 | 25 | P7 | 26 | P7 | 27 | P7 |
28 | P7 | 29 | P7 | 30 | P7 | 31 | P7 | 32 | P7 | 33 | P7 | 34 | P7 | 35 | P7 | 36 | P7 | 37
| P7 | 38 |

```