30/5/24

PAGE NO.:
DATE / /

1) Write a C program to stimulate the following CPU scheduling algorithm to find TAT & WT.

a) SJF (Pre emptive)

```
#include <stdio.h>
void main(){
    int a[10], b[10], x[10];
    int w[10], TA[10], c[10];
    int i, j, small, cnt=0, t, n;
    double avg=0, tt=0, end;
    printf("Enter nor of processes: \n");
    scanf("%d", &n);
    for(i=0; i<n; i++){
        printf("Enter arrival time \n");
        scanf("%d", &a[i]);
    }
    for(i=0; i<n; i++){
        printf("Enter burst time \n");
        scanf("%d", &b[i]);
    }
    for(i=0; i<n; i++)
        x[i]=b[i]
    b[9]=9999;
    for(t=0; cnt!=0; t++){
        smallest=9;
        for(i=0; i<n; i++){
            if(a[i]<=t && b[i]<b[small] && b[i]>0)
            small=i;
        }
        b[small]--;
        if(b[small]==0){
            cnt++;
            end=t+1;
            c[small]=end;
            w[small]=end-a[small]-x[small];
```

```
    TA[small] = end - a[small];
  }}
printf("pid \t burst \t arrival \t waiting \t TAT \t Completion");
for(i=0; i<n; i++){
    printf("%d \t %d \t %d \t %d \t %d \t %d \t", i+1,
        x[i], a[i], w[i], TA[i], c[i]);
    avg = avg + w[i];
    tt = tt + TA[i];
}
printf("Avg WT \n", avg/n);
printf("Avg TAT \n", tt/n);
}
```

→ OUTPUT:-

Enter nor of process = 4
Enter AT of P1 = 0
Enter AT of P2 = 1
Enter AT of P3 = 2
Enter AT of P4 = 3

Enter BT of P1 = 8
Enter BT of P2 = 4
Enter BT of P3 = 9
Enter BT of P4 = 5

Pid	burst	Arrival	Waiting	TAT	Completion
1	8	0	9	17	17
2	4	1	0	4	5
3	9	2	15	24	26
4	5	3	2	7	10

Avg WT = 6.500
Avg TAT = 13.000

6) Priority (Pre-emptive):

```
#include <stdio.h>
#define MAX 9999;
struct proc{
    int n, at, bt, rt, ct, wt, tat, pri, temp;
};
struct proc read(int i){
    struct proc p;
    printf("Process 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.rt = p.bt;
    p.temp = p.pri;
    return p;
}
void main(){
    int i, n, c, rem, min, min_i;
    struct proc p[10], temp;
    float avg tat=0, avgwt=0;
    printf("Enter nos of processes:");
    scanf("%d", &n);
    for(i=0; i<n; i++)
        p[i] = read(i+1);
    remaining = n;
    for(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;
    }
    min = p[0].temp, min_i = 0;
    for (j = 0; j<n && p[j].at <= p[0].at; j++)
        if (p[j].temp < min)
            min = p[j].temp, min_i = j;
    i = min_i;
    p[i].ct = p[i].at + 1;
    p[i].rt--;
    if (p[i].rt == 0){
        p[i].temp = MAX;
        rem--;
    }
    while (rem > 0){
        min = p[0].temp, min_i = 0;
        for (j=0; j<n && p[j].at <= c; j++)
            if (p[j].temp < min)
                min = p[j].temp, min_i = j;
        i = min_i;
        p[i].ct = c + 1;
        p[i].rt--;
        if (p[i].rt == 0){
            p[i].temp = MAX;
            rem--;
        }
    }
    printf("\n Process no \t AT \t BT \t pri \t CT \t TAT \t WT \n");
    for (i=0; i<n; i++){
        p[i].tat = p[i].ct - p[i].at;
        avg.tat += p[i].tat;
        p[i].wt = p[i].tat - p[i].bt;
        avg wt += p[i].wt;
```

```
printf("%d\t%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("Avg TAT = %f \n Avg WT = %f", avgtat, avgwt);
}
```

→ OUTPUT:- Enter nor of process : 5

Process no: 1
AT: 0 BT: 3 Pr: 5
Pro no: 2
AT: 2 BT: 2 Pr: 3
Pro no: 3
AT: 3 BT: 5 Pr: 2
Pro no: 4
AT: 4 BT: 4 Pr: 4
Pro no: 5
AT: 6 BT: 1 Pr: 1

Process No	AT	BT	Pri	CT	TAT	WT
P1	0	3	5	15	15	12
P2	2	2	3	10	8	6
P3	3	5	2	9	6	1
P4	4	4	4	14	10	6
P5	5	1	1	7	1	0

Avg TAT = 8.000
Avg WT = 5.000

c) Priority (Non-preemptive):

```
#include <stdio.h>
#define MAX 9999;
struct proc{
    int n, at, bt, ct, wt, tat, pri, sts;
};
struct proc read(int i){
    struct proc p;
    printf("Process no: %d\n", i);
    p.no = i;
    printf("Enter AT: ");
    scanf("%d", &p.at);
    printf("Enter BT: ");
    scanf("%d", &p.bt);
    printf("Enter priority");
    scanf("%d", &p.pri);
    p.sts = 0;
    return p;
}
void main(){
    int n, s, ct=0, sum, i, j;
    struct proc p[10], temp;
    float avgtat=0, avgwt=0;
    printf("Enter nos of process\n");
    scanf("%d", &n);
    for(i=0; i<n; i++)
        p[i] = read(i+1);
    for(i=0; i<n-1; i++)
        for(j=0; j<n-1-i; j++)
            if(p[j].at > p[j+1].at){
                temp = p[j];
                p[j] = p[j+1];
                p[j+1] = temp;
            }
```

```
p[a].pri=MAX;
rem=n;
printf("\n Process \t AT \t BT \t Pri \t CT \t TAT \t WT");
for(ct=p[0].at; rem!=0;){
    s=a;
    for(i=0;i<n;i++)
        if(p[i].at<=ct && p[i].sts!=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].sts=1;
    rem--;
    printf("P%d\t%d\t%d\t%d\t%d\t%d\t%d
        \t", p[s].no, p[s].at, p[s].bt, p[s].pri,
        p[s].ct, p[s].tat, p[s].wt);
}
avgtat/=n; avgwt/=n;
printf("Avg TAT:%f \n AvgWT=%f", avgtat, avgwt);
}
```

PAGE NO.:
DATE / /

→ Output:

Enter number of process : 5

Process No: 1
AT: 0 BT: 3 Pr: 5

Process No: 2
AT: 2 BT: 2 Pr: 3

Process No: 3
AT: 3 BT: 5 Pr: 2

Process No: 4
AT: 4 BT: 4 Pr: 4

Process No: 5
AT: 6 BT: 1 Pr: 1

Process No	AT	BT	Pri	CT	TAT	WT
P1	0	3	5	3	3	0
P3	3	5	2	8	5	0
P5	6	1	1	9	3	2
P2	2	2	3	11	9	7
P4	4	4	4	15	11	7

Avg TAT = 6.200000
Avg WT = 3.200000

Sam
3/11/25

PAGE NO.:
DATE / /

d) round robin:

```
#include <stdio.h>
struct proc {
    int n, at, bt, ct, tat, wt, rl;
};
struct proc read(int i){
    struct proc p;
    printf("Process No: %d\n", i);
    p.no = i;
    printf("Enter AT\n");
    scanf("%d", &p.at);
    printf("Enter BT\n");
    scanf("%d", &p.bt);
    p.rt = p.bt;
    return p;
}
int main(){
    struct proc p[10], tmp;
    float avgtat=0, avgwt=0;
    int n, tq, ct=0, flag=0, rem;
    printf("Enter no of processes\n");
    scanf("%d", &n);
    printf("Enter Time Quantum\n");
    scanf("%d", &tq);
    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;
            }
```

```
rem=n;
printf("\n Process \t AT \t BT \t CT \t TAT \t WT\n");
for(int i=0; rem!=0;){
    if(p[i].rt<=tq && p[i].rt>0){
        ct+=p[i].rt;
        p[i].rt=0;
        flag=1;
    }
    else if(p[i].rt>0){
        p[i].rt-=tq;
        ct+=tq;
    }
    if(p[i].rt==0 && flag==1){
        flag=0;
        rem--;
        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\t",
            p[i].no, p[i].at, p[i].bt, p[i].ct, p[i].tat,
            p[i].wt);
    }
    if(i!=n-1 && p[i+1].at<=ct)
        i++;
    else
        i=0;
}
avgtat/=n, avgwt/=n;
printf("Avg TAT: %f, Avg WT: %f", avgtat,
    avgwt);
}
```

PAGE NO.:
DATE / /

→ Output :

Enter no of processes = 4
Enter Time Quantum = 5
Process no: 1
AT: 0
BT: 21
Process no: 2
AT: 0
BT: 3
Process no: 3
AT: 0
BT: 6
Process no: 4
AT: 0
BT: 2

Process No	AT	BT	CT	TAT	WT
P2	0	3	8	8	5
P4	0	2	15	15	13
P3	0	6	21	21	15
P1	0	21	32	32	11

Avg TAT : 19.000
Avg WT : 11.000

Aeu
31/5/24