

LAB – 04

Aim : Write a program to simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time.

- a) FCFS b) SJF

a) FCFS

First come first serve(FCFS) scheduling algorithm simply schedules the jobs according to their arrival time. The job which comes first in the ready queue will get the CPU first.

Program :

```
#include<stdio.h>

int main()
{
int p[10],at[10],bt[10],ct[10],tat[10],wt[10],i,j,temp=0,n;
float awt=0,atat=0;
printf("enter no of process you want:");
scanf("%d",&n);
printf("enter %d process:",n);
for(i=0;i<n;i++){
scanf("%d",&p[i]);
}
printf("enter %d arrival time:",n);
for(i=0;i<n;i++)
{
scanf("%d",&at[i]);
}
printf("enter %d burst time:",n);
```

```

for(i=0;i<n;i++){
scanf("%d",&bt[i]);
}
for(i=0;i<n;i++)
{
for(j=0;j<(n-i);j++){
if(at[j]>at[j+1])
{
temp=p[j+1];
p[j+1]=p[j];
p[j]=temp;
temp=at[j+1];
at[j+1]=at[j];
at[j]=temp;
temp=bt[j+1];
bt[j+1]=bt[j];
bt[j]=temp;
} } }
ct[0]=at[0]+bt[0];
for(i=1;i<=n;i++)
{
temp=0;
if(ct[i-1]<at[i])
{
temp=at[i]-ct[i-1];
}
ct[i]=ct[i-1]+bt[i]+temp;

```

```

}

printf("\n p\t A.T\t B.T\t C.T\t TAT\t WT");

for(i=1;i<=n;i++){

tat[i]=ct[i]-at[i];

wt[i]=tat[i]-bt[i];

atat+=tat[i];

awt+=wt[i];

}

printf("%d",tat[n-1]);

atat=atat/n;

awt=awt/n;

for(i=1;i<=n;i++)

{

printf("\nP%d\t %d\t %d\t %d \t %d \t %d",p[i],at[i],bt[i],ct[i],tat[i],wt[i]);

}

printf("\naverage turnaround time is %f",atat);

printf("\naverage waiting time is %f",awt);

return 0; }

```

Output :

```

enter the number of process:4
enter the process1 2 3 4
enter the arrival time0 2 2 0
enter the burst time5 3 1 3

p      at      bt      ct      tat      wt
4      0      3      3      3      0
3      2      1      4      2      1
2      2      3      7      5      2
1      0      5      12     12      7

average turn around time=5.50 and average waiting time=2.50
-----
Process exited after 36.66 seconds with return value 0
Press any key to continue . . .

```

b) SJF

The shortest job first (SJF) or shortest job next, is a scheduling policy that selects the waiting process with the smallest execution time to execute next. SJN, also known as Shortest Job Next (SJN).

Program :

```
#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

void swap(int *x, int *y)

{

int temp=*x;

*x=*y;

*y=temp;

}

void sortat(int p[], int at[], int bt[], int n)

{

int i, j;

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

{

for(j=i+1;j<n;j++)

{

if(at[i]>at[j])

{

swap(&p[i], &p[j]);

swap(&at[i], &at[j]);

swap(&bt[i], &bt[j]);

}

}

}
```

```

else if(at[i]==at[j])
{
if(bt[i]>bt[j])
swap(&p[i], &p[j]);
swap(&at[i], &at[j]);
swap(&bt[i], &bt[j]);
} } } }

void tatwt( int ct[], int at[], int bt[], int tat[], int wt[], int n)
{
int i;
for(i=0;i<n;i++)
{
tat[i]=ct[i]-at[i];
wt[i]=tat[i]-bt[i];
} }

int main()
{
int *p, *at, *bt, *tat, *wt, *ct, pos, i, j, min=1000, n;
float awt=0, atat=0;
printf("\nenter the number of process:");
scanf("%d", &n);
p=(int*)malloc(n*sizeof(int));
at=(int*)malloc(n*sizeof(int));
bt=(int*)malloc(n*sizeof(int));
ct=(int*)malloc(n*sizeof(int));
wt=(int*)malloc(n*sizeof(int));
tat=(int*)malloc(n*sizeof(int));

```

```
printf("enter the process");
for(i=0;i<n;i++){
scanf("%d",&p[i]);
}
printf("enter the arrival time");
for(i=0;i<n;i++) {
scanf("%d",&at[i]);
}
printf("enter the burst time");
for(i=0;i<n;i++) {
scanf("%d",&bt[i]);
}
sortat(p, at, bt, n);
ct[0]=at[0] + bt[0];
for(i=1; i<n; i++){
for(j=i; j<n; j++){
if(at[j]<=ct[i-1])
{
if(bt[j]<min)
{
min=bt[j];
pos=j;
} } }
swap(&p[i], &p[pos]);
swap(&at[i], &at[pos]);
swap(&bt[i], &bt[pos]);
min=1000;
```

```

ct[i]=ct[i-1]+bt[i];
}
tatwt(ct, at, bt, tat, wt, n);
printf("\n p\t at\t bt\t ct\t tat\t wt");
for(i=0;i<n;i++){
printf("\n %d\t %d\t %d\t %d\t %d\t %d",p[i], at[i], bt[i], ct[i], tat[i], wt[i]);
}
for(i=0;i<n;i++){
atat+=tat[i];
awt+=wt[i];
}
atat=atat/n;
awt=awt/n;
printf("\n average turn around time=%.2f and average waiting time=%.2f",atat,
awt);
return 0; }

```

Output :

```

enter no of process you want:5
enter 5 process:1 2 3 4 5
enter 5 arrival time:5 3 4 1 2
enter 5 burst time:5 4 4 3 7

p      A.T      B.T      C.T      TAT      WT15
P4      1        3        4        3        0
P5      2        7       11        9        2
P2      3        4       15       12        8
P3      4        4       19       15       11
P1      5        5       24       19       14
average turnaround time is 11.600000
average waiting time is 7.000000
-----
Process exited after 20.21 seconds with return value 0
Press any key to continue . . .

```