Naming the file: 20BCE1798_EX-4_CPU Scheduling Algorithms.

Reg No: **20BCE1798**
Name: **ANSH GOEL**

Course Code: CSE2005
Course Name: Operating Systems (Embedded Lab)

Slot: L27+L28

Ex No 4: CPU Scheduling Algorithms
Date: 04-02-2022

1. FCFS (Non -Pre-emptive)

CODE:

```
#include<stdio.h>
int main()
{
int n,bt[20],wt[20],tat[20],avwt=0,avtat=0,i,j;
printf("Enter total number of processes(maximum 20):");
scanf("%d",&n);
printf("\nEnter Process Burst Timen");
for(i=0;i<n;i++)
{
printf("\nP[%d]:",i+1);
scanf("%d",&bt[i]);
}
wt[0]=0;
for(i=1;i<n;i++)
{
wt[i]=0;
for(j=0;j<i;j++)
wt[i]+=bt[j];
}
printf("\nProcessttBurst TimetWaiting TimetTurnaround Time");
for(i=0;i<n;i++)
{
tat[i]=bt[i]+wt[i];
avwt+=wt[i];
avtat+=tat[i];
printf("\nP[%d] tt%d tt%d tt%d",i+1,bt[i],wt[i],tat[i]);
}
avwt/=i;
avtat/=i;
printf("\nAverage Waiting Time:%d",avwt);
printf("\nAverage Turnaround Time:%d",avtat);
return 0;
}
```

```c
#include<stdio.h>
int main()
{
int n,bt[20],wt[20],tat[20],avwt=0,avtat=0,i,j;
printf("Enter total number of processes(maximum 20):");
scanf("%d",&n);
printf("\nEnter Process Burst Timen");
for(i=0;i<n;i++)
{
printf("\nP[%d]:",i+1);
scanf("%d",&bt[i]);
}
wt[0]=0;
for(i=1;i<n;i++)
{
wt[i]=0;
for(j=0;j<i;j++)
wt[i]+=bt[j];
}
printf("\nProcessttBurst TimetWaiting TimetTurnaround Time");
for(i=0;i<n;i++)
{
tat[i]=bt[i]+wt[i];
avwt+=wt[i];
avtat+=tat[i];
printf("\nP[%d] tt%d tt%d tt%d",i+1,bt[i],wt[i],tat[i]);
}
avwt/=i;
avtat/=i;
printf("\nAverage Waiting Time:%d",avwt);
printf("\nAverage Turnaround Time:%d",avtat);
return 0;
}
```

OUTPUT:

```
Enter total number of processes(maximum 20):4

Enter Process Burst Timen
P[1]:2

P[2]:5

P[3]:1

P[4]:7

ProcessttBurst TimetWaiting TimetTurnaround Time
P[1] tt2 tt0 tt2
P[2] tt5 tt2 tt7
P[3] tt1 tt7 tt8
P[4] tt7 tt8 tt15
Average Waiting Time:4
Average Turnaround Time:8

...Program finished with exit code 0
Press ENTER to exit console.
```

NAME – ANSH GOEL REG NO.- 20BCE1798

2. **SJF (Shortest Job First)** falls under non-pre-emptive category

CODE:

```
#include<stdio.h>
int main()
{
int bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp;
float avg_wt,avg_tat;
printf("Enter number of process:");
scanf("%d",&n);
printf("\nProcessId : Enter Burst Time");
for(i=0;i<n;i++)
{
printf("\nP%d : ",i+1);
scanf("%d",&bt[i]);
p[i]=i+1;
}
//sorting of burst times
for(i=0;i<n;i++)
{
pos=i;
for(j=i+1;j<n;j++)
{
if(bt[j]<bt[pos])
pos=j;
}
temp=bt[i];
bt[i]=bt[pos];
bt[pos]=temp;
temp=p[i];
p[i]=p[pos];
p[pos]=temp;
}
wt[0]=0;
for(i=1;i<n;i++)
{
wt[i]=0;
for(j=0;j<i;j++)
wt[i]+=bt[j];
total+=wt[i];
}
avg_wt=(float)total/n;
total=0;
printf("\nProcesst Burst Time tWaiting TimetTurnaround Time");
for(i=0;i<n;i++)
{
tat[i]=bt[i]+wt[i];
total+=tat[i];
printf("\np%d tt%d tt%d tt%d",p[i],bt[i],wt[i],tat[i]);
}
avg_tat=(float)total/n;
printf("\nAverage Waiting Time=%f",avg_wt);
printf("\nAverage Turnaround Time=%f",avg_tat);
}
```

```c
#include<stdio.h>
int main()
{
int bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp;
float avg_wt,avg_tat;
printf("Enter number of process:");
scanf("%d",&n);
printf("\nProcessId : Enter Burst Time");
for(i=0;i<n;i++)
{
printf("\nP%d : ",i+1);
scanf("%d",&bt[i]);
p[i]=i+1;
}
//sorting of burst times
for(i=0;i<n;i++)
{
pos=i;
for(j=i+1;j<n;j++)
{
if(bt[j]<bt[pos])
pos=j;
}
temp=bt[i];
bt[i]=bt[pos];
bt[pos]=temp;
temp=p[i];
p[i]=p[pos];
p[pos]=temp;
}
wt[0]=0;
for(i=1;i<n;i++)
{
wt[i]=0;
for(j=0;j<i;j++)
wt[i]+=bt[j];
total+=wt[i];
}
avg_wt=(float)total/n;
total=0;
printf("\nProcesst Burst Time tWaiting TimetTurnaround Time");
for(i=0;i<n;i++)
{
tat[i]=bt[i]+wt[i];
total+=tat[i];
printf("\np%d tt%d tt%d tt%d",p[i],bt[i],wt[i],tat[i]);
}
avg_tat=(float)total/n;
printf("\nAverage Waiting Time=%f",avg_wt);
printf("\nAverage Turnaround Time=%f",avg_tat);
}
```

OUTPUT:

```
Enter number of process:4

ProcessId : Enter Burst Time
P1 : 2

P2 : 1

P3 : 5

P4 : 7

Processt Burst Time tWaiting TimetTurnaround Time
p2 tt1 tt0 tt1
p1 tt2 tt1 tt3
p3 tt5 tt3 tt8
p4 tt7 tt8 tt15
Average Waiting Time=3.000000
Average Turnaround Time=6.750000

...Program finished with exit code 0
Press ENTER to exit console.
```

NAME – ANSH GOEL REG NO.- 20BCE1798

3. **<u>SRTF (Shortest Remaining Time First)</u>**, another name of SJF & falls under Pre-emptive Category.

CODE:

```
#include <stdio.h>
int main()
{
int arrival_time[10], burst_time[10], temp[10];
int i, smallest, count = 0, time, limit;
double wait_time = 0, turnaround_time = 0, end;
float average_waiting_time, average_turnaround_time;
printf("Enter the Total Number of Processes:");
scanf("%d", &limit);
printf("\nEnter Details of %d Processes", limit);
for(i = 0; i < limit; i++)
{
printf("\nEnter Arrival Time:");
scanf("%d", &arrival_time[i]);
printf("Enter Burst Time:");
scanf("%d", &burst_time[i]);
temp[i] = burst_time[i];
}
burst_time[9] = 9999;
for(time = 0; count != limit; time++)
{
smallest = 9;
for(i = 0; i < limit; i++)
{
if(arrival_time[i] <= time && burst_time[i] < burst_time[smallest] && burst_time[i] > 0)
{
smallest = i;
}
}
burst_time[smallest]--;
if(burst_time[smallest] == 0)
{
count++;
end = time + 1;
wait_time = wait_time + end - arrival_time[smallest] - temp[smallest];
turnaround_time = turnaround_time + end - arrival_time[smallest];
}
}
average_waiting_time = wait_time / limit;
average_turnaround_time = turnaround_time / limit;
printf("\nAverage Waiting Time: %lf", average_waiting_time);
printf("\nAverage Turnaround Time: %lf", average_turnaround_time);
return 0;
}
```

```c
#include <stdio.h>
int main()
{
int arrival_time[10], burst_time[10], temp[10];
int i, smallest, count = 0, time, limit;
double wait_time = 0, turnaround_time = 0, end;
float average_waiting_time, average_turnaround_time;
printf("Enter the Total Number of Processes:");
scanf("%d", &limit);
printf("\nEnter Details of %d Processes", limit);
for(i = 0; i < limit; i++)
{
printf("\nEnter Arrival Time:");
scanf("%d", &arrival_time[i]);
printf("Enter Burst Time:");
scanf("%d", &burst_time[i]);
temp[i] = burst_time[i];
}
burst_time[9] = 9999;
for(time = 0; count != limit; time++)
{
smallest = 9;
for(i = 0; i < limit; i++)
{
if(arrival_time[i] <= time && burst_time[i] < burst_time[smallest] && burst_time[i] > 0)
{
smallest = i;
}
}
burst_time[smallest]--;
if(burst_time[smallest] == 0)
{
count++;
end = time + 1;
wait_time = wait_time + end - arrival_time[smallest] - temp[smallest];
turnaround_time = turnaround_time + end - arrival_time[smallest];
}
}
average_waiting_time = wait_time / limit;
average_turnaround_time = turnaround_time / limit;
printf("\nAverage Waiting Time: %lf", average_waiting_time);
printf("\nAverage Turnaround Time: %lf", average_turnaround_time);
return 0;
}
```

OUTPUT:

```
Enter the Total Number of Processes:4

Enter Details of 4 Processes
Enter Arrival Time:5
Enter Burst Time:2

Enter Arrival Time:9
Enter Burst Time:5

Enter Arrival Time:0
Enter Burst Time:1

Enter Arrival Time:1
Enter Burst Time:7

Average Waiting Time: 0.750000
Average Turnaround Time: 4.500000

...Program finished with exit code 0
Press ENTER to exit console.
```

NAME – ANSH GOEL REG NO.- 20BCE1798