**OS LAB ASSIGNMENT 6**

1. Preemitive SJF

#include<stdio.h>

#include<conio.h>

#include<string.h>

int main()

{

int et[20],at[10],n,i,j,temp,st[10],ft[10],wt[10],ta[10];

int totwt=0,totta=0;

float awt,ata;

char pn[10][10],t[10];

//clrscr();

printf("Enter the number of process:");

scanf("%d",&n);

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

{

printf("Enter process name, arrival time& execution time:");

//flushall();

scanf("%s%d%d",pn[i],&at[i],&et[i]);

}

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

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

{

if(et[i]<et[j])

{

temp=at[i];

at[i]=at[j];

at[j]=temp;

temp=et[i];

et[i]=et[j];

et[j]=temp;

strcpy(t,pn[i]);

strcpy(pn[i],pn[j]);

strcpy(pn[j],t);

}

}

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

{

if(i==0)

st[i]=at[i];

else

st[i]=ft[i-1];

wt[i]=st[i]-at[i];

ft[i]=st[i]+et[i];

ta[i]=ft[i]-at[i];

totwt+=wt[i];

totta+=ta[i];

}

awt=(float)totwt/n;

ata=(float)totta/n;

printf("\nPname\tarrivaltime\texecutiontime\twaitingtime\ttatime");

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

printf("\n%s\t%5d\t\t%5d\t\t%5d\t\t%5d",pn[i],at[i],et[i],wt[i],ta[i]);

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

printf("\nAverage turnaroundtime is:%f",ata);

getch();

}

Output

Enter the number of process:4

Enter process name, arrival time& execution time:1

2

4

Enter process name, arrival time& execution time:2

6

5

Enter process name, arrival time& execution time:3

8

5

Enter process name, arrival time& execution time:4

3

6

Pname arrivaltime executiontime waitingtime tatime

1 2 4 0 4

3 8 5 -2 3

2 6 5 5 10

4 3 6 13 19

Average waiting time is:4.000000

Average turnaroundtime is:9.000000

2)SJF Non Preemitive

#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("\nEnter Burst Time:\n");

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

{

printf("p%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("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time");

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

{

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

total+=tat[i];

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

}

avg\_tat=(float)total/n;

printf("\n\nAverage Waiting Time=%f",avg\_wt);

printf("\nAverage Turnaround Time=%f\n",avg\_tat);

}

Output

Enter number of process:4

Enter Burst Time:

p1:2

p2:3

p3:4

p4:5

Process Burst Time Waiting Time Turnaround Time

p1 2 0 2

p2 3 2 5

p3 4 5 9

p4 5 9 14

Average Waiting Time=4.000000

Average Turnaround Time=7.500000

3)Priority

#include<stdio.h>

#include<conio.h>

#include<string.h>

int main()

{

int et[20],at[10],n,i,j,temp,p[10],st[10],ft[10],wt[10],ta[10];

int totwt=0,totta=0;

float awt,ata;

char pn[10][10],t[10];

printf("Enter the number of process:");

scanf("%d",&n);

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

{

printf("Enter process name,arrivaltime,execution time & priority:");

scanf("%s%d%d%d",pn[i],&at[i],&et[i],&p[i]);

}

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

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

{

if(p[i]<p[j])

{

temp=p[i];

p[i]=p[j];

p[j]=temp;

temp=at[i];

at[i]=at[j];

at[j]=temp;

temp=et[i];

et[i]=et[j];

et[j]=temp;

strcpy(t,pn[i]);

strcpy(pn[i],pn[j]);

strcpy(pn[j],t);

}

}

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

{

if(i==0)

{

st[i]=at[i];

wt[i]=st[i]-at[i];

ft[i]=st[i]+et[i];

ta[i]=ft[i]-at[i];

}

else

{

st[i]=ft[i-1];

wt[i]=st[i]-at[i];

ft[i]=st[i]+et[i];

ta[i]=ft[i]-at[i];

}

totwt+=wt[i];

totta+=ta[i];

}

awt=(float)totwt/n;

ata=(float)totta/n;

printf("\nPname\tarrivaltime\texecutiontime\tpriority\twaitingtime\ttatime");

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

printf("\n%s\t%5d\t\t%5d\t\t%5d\t\t%5d\t\t%5d",pn[i],at[i],et[i],p[i],wt[i],ta[i]);

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

printf("\nAverage turnaroundtime is:%f",ata);

getch();

}

Output

Enter the number of process:4

Enter process name,arrivaltime,execution time & priority:1

3

4

2

Enter process name,arrivaltime,execution time & priority:2

4

5

1

Enter process name,arrivaltime,execution time & priority:3

4

5

3

Enter process name,arrivaltime,execution time & priority:4

5

7

4

Pname arrivaltime executiontime priority waitingtime tatime

2 4 5 1 0 5

1 3 4 2 6 10

3 4 5 3 9 14

4 5 7 4 13 20

Average waiting time is:7.000000

Average turnaroundtime is:12.250000

4)Round Robin

#include<stdio.h>

int main()

{

int i, limit, total = 0, x, counter = 0, time\_quantum;

int wait\_time = 0, turnaround\_time = 0, arrival\_time[10], burst\_time [10], temp[10];

float average\_wait\_time, average\_turnaround\_time;

printf("\nEnter Total Number of Processes: ");scanf("%d", &limit);

x = limit;

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

{

printf("\nEnter Details of Process[%d]\n", i + 1);

printf("Arrival Time: ");

scanf("%d", &arrival\_time[i]);

printf("Burst Time: ");

scanf("%d", &burst\_time[i]);

temp[i] = burst\_time[i];

}

printf("\nEnter Time Quantum: ");

scanf("%d", &time\_quantum);

printf("\nProcess ID\t\tBurst Time\t Turnaround Time\t Waiting Time\n"); for (total = 0, i

= 0; x != 0;)

{

if (temp[i] <= time\_quantum && temp[i] > 0)

{

total = total + temp[i];

temp[i] = 0;

counter = 1;

}

else if (temp[i] > 0)

{

temp[i] = temp[i] - time\_quantum;

total = total + time\_quantum;

}

if (temp[i] == 0 && counter == 1)

{

x--;

printf("\nProcess[%d]\t\t%d\t\t %d\t\t\t %d", i + 1, burst\_time[ i], total - arrival\_time[i], total -

arrival\_time[i] - burst\_time[i]);

wait\_time = wait\_time + total - arrival\_time[i] -

burst\_time[i];

turnaround\_time = turnaround\_time + total - arrival\_time[i];

counter = 0;

}

if (i == limit - 1)

{

i = 0;

}

else if (arrival\_time[i + 1] <= total)

{

i++;

}

else

{

i = 0;

}

}

average\_wait\_time = wait\_time \* 1.0 / limit;

average\_turnaround\_time = turnaround\_time \* 1.0 / limit;

printf("\n\nAverage Waiting Time:\t%f", average\_wait\_time);

printf("\nAvg Turnaround Time:\t%f\n", average\_turnaround\_time);

return 0;

}

Output

Enter Total Number of Processes: 3

Enter Details of Process[1]

Arrival Time: 5

Burst Time: 8

Enter Details of Process[2]

Arrival Time: 7

Burst Time: 2

Enter Details of Process[3]

Arrival Time: 1

Burst Time: 6

Enter Time Quantum: 1

Process ID Burst Time Turnaround Time Waiting Time

Process[1] 8 5 -3

Process[2] 2 4 2

Process[3] 6 15 9

Average Waiting Time: 2.666667

Avg Turnaround Time: 8.000000