Tutorial Number:

TITLE:Scheduling

NAME: Poonam Maruti Bagate ROLLNO:04

CLASS: T.E. BRANCH: Comp BATCH: E-1

DATE OF PERFORMANCE:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include<stdio.h>

void FCFS();

void SJF();

void nonPrio();

void RR();

void main()

{

int choice;

do

{

printf("\nEnter your choice\n");

printf("1.FCFS\n2.SJF(premptive)\n3.Priority(non-premptive)\n4.Round Robin\n5.quit");

scanf("%d",&choice);

switch(choice)

{

case 1:

FCFS();

break;

case 2:

SJF();

break;

case 3:nonPrio();

break;

case 4:RR();

break;

}

}while(choice!=5);

}

void FCFS()

{

int i,pro,arr[11],bur[11],fin[11];

printf("Enter the number of processes\n");

scanf("%d",&pro);

printf("Enter the arrival times\n");

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

scanf("%d",&arr[i]);

printf("Enter the burst times\n");

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

scanf("%d",&bur[i]);

fin[0]=arr[0]+bur[0];

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

fin[i]=fin[i-1]+bur[i]; //caluculating fininsh time of each process

printf("\nP.Id\tArival\tBurst\tFinish\tTurn\tWaiting\n");

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

printf("P%d\t%d\t%d\t%d\t%d\t%d\n",i+1,arr[i],bur[i],fin[i],fin[i]-arr[i],fin[i]-arr[i]-bur[i]);

printf("Gantt Chart");

printf("\n-----------------------------------------------------\n");

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

{

printf("P\_id|%d|FT|%d|",arr[i],fin[i]);

}

printf("\n--------------------------------------------------------\n");

}

void SJF()

{

int i,k,flag,min=999,pro,arr[11],bur[11],fin[11],status[11]={0},T=0;

printf("Enter the number of processes\n");

scanf("%d",&pro);

printf("Enter the arrival times\n");

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

scanf("%d",&arr[i]);

printf("Enter the burst times\n");

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

scanf("%d",&bur[i]);

while(1)

{

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

{

if(arr[i]<=T&&(status[i]!=-1||status[i]==0))

{

status[i]=1;

}

}

min=999;

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

{

if(status[i]==1&&bur[i]<min)

{

min=bur[i];k=i;

}

}

fin[k]=T+bur[k];status[k]=-1;

T+=bur[k]; flag=0;

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

if(status[i]==-1)

flag=1;

else

{flag=0;break;}

if(flag==1)break;

}

printf("\nP.Id\tArival\tBurst\tFinish\tTurn\tWaiting\n");

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

printf("P%d\t%d\t%d\t%d\t%d\t%d\n",i+1,arr[i],bur[i],fin[i],fin[i]-arr[i],fin[i]-arr[i]-bur[i]);

printf("Gantt Chart");

printf("\n-----------------------------------------------------\n");

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

{

printf("P\_id|%d|FT|%d|",arr[i],fin[i]);

}

printf("\n--------------------------------------------------------\n");

}

void nonPrio()

{

int i,k,flag,min=999,pro,arr[11],bur[11],pri[11],fin[11],status[11]={0},T=0;

printf("Enter the number of processes\n");

scanf("%d",&pro);

printf("Enter the arrival times\n");

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

scanf("%d",&arr[i]);

printf("Enter the burst times\n");

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

scanf("%d",&bur[i]);

printf("Enter the priorities\n");

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

scanf("%d",&pri[i]);

while(1)

{

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

{

if(arr[i]<=T&&(status[i]!=-1||status[i]==0))

{

status[i]=1;

}

}

min=999;

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

{

if(status[i]==1&&pri[i]<min)

{min=pri[i];k=i;}

}

fin[k]=T+bur[k];status[k]=-1;

T+=bur[k]; flag=0;

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

if(status[i]==-1)

flag=1;

else

{flag=0;break;}

if(flag==1)break;

}

printf("\nP.Id\tArrival\tBurst\tPrio\tFinish\tTurn\tWaiting\n");

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

printf("P%d\t%d\t%d\t%d\t%d\t%d\t%d\n",i+1,arr[i],bur[i],pri[i],fin[i],fin[i]-arr[i],fin[i]-arr[i]-bur[i]);

printf("Gantt Chart");

printf("\n-----------------------------------------------------\n");

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

{

printf("P\_id|%d|FT|%d|",arr[i],fin[i]);

}

printf("\n--------------------------------------------------------\n");

}

void RR()

{

int tp;

int i,k,swp,j=-1,flag,s,t,pro,arr[11],bur[11],que[11],temp[11],fin[11],status[11]={0},T=0;

printf("Enter the number of processes\n");

scanf("%d",&pro);

printf("Enter the arrival times\n");

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

scanf("%d",&arr[i]);

printf("Enter the burst times\n");

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

{scanf("%d",&bur[i]);temp[i]=bur[i];}

printf("Enter the time slab\n");

scanf("%d",&t);

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

{

if(arr[i]==T)

{ j++;

status[i]=1;que[j]=i;

}

}

while(1)

{

if(bur[que[0]]>t)

{

swp=1;

for(tp=0;tp<t;tp++)

{

bur[que[0]]--; T++;

for(k=0;k<pro;k++)

{

if(arr[k]==T&&status[k]==0)

{

j++;

que[j]=k; status[k]=1;

}

}

}

}

else

{

swp=0;

while(bur[que[0]]!=0)

{

bur[que[0]]--; T++;

for(k=0;k<pro;k++)

if(arr[k]==T&&status[k]==0)

{

j++;

que[j]=k; status[k]=1;

}

}

fin[que[0]]=T;

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

{

que[k]=que[k+1];

}

j--;

}

if(swp==1)

{

s=que[0];

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

{

que[k]=que[k+1];

}

que[j]=s;

}

for(k=0;k<pro;k++)

{

if(bur[k]==0)

flag=1;

else

{flag=0;break;}

}

if(flag==1)break;

}

printf("\nP.Id\tArival\tBurst\tFinish\tTurn\tWaiting\n");

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

printf("P%d\t%d\t%d\t%d\t%d\t%d\n",i+1,arr[i],temp[i],fin[i],fin[i]-arr[i],fin[i]-arr[i]-temp[i]);

printf("Gantt Chart");

printf("\n-----------------------------------------------------\n");

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

{

printf("P\_id|%d|FT|%d|",arr[i],fin[i]);

}

printf("\n--------------------------------------------------------\n");

}

/\*poonam@ubuntu:~/Desktop$ gcc schedul.c

poonam@ubuntu:~/Desktop$ ./a.out

Enter your choice

1.FCFS

2.SJF(premptive)

3.Priority(non-premptive)

4.Round Robin

5.quit1

Enter the number of processes

5

Enter the arrival times

0

1

3

9

12

Enter the burst times

3

5

2

5

5

P.Id Arival Burst Finish Turn Waiting

P1 0 3 3 3 0

P2 1 5 8 7 2

P3 3 2 10 7 5

P4 9 5 15 6 1

P5 12 5 20 8 3

Gantt Chart

-----------------------------------------------------

P\_id|0|FT|3|P\_id|1|FT|8|P\_id|3|FT|10|P\_id|9|FT|15|P\_id|12|FT|20|

--------------------------------------------------------

Enter your choice

1.FCFS

2.SJF(premptive)

3.Priority(non-premptive)

4.Round Robin

5.quit2

Enter the number of processes

5

Enter the arrival times

0

1

3

9

12

Enter the burst times

3

5

2

5

5

P.Id Arival Burst Finish Turn Waiting

P1 0 3 3 3 0

P2 1 5 10 9 4

P3 3 2 5 2 0

P4 9 5 15 6 1

P5 12 5 20 8 3

Gantt Chart

-----------------------------------------------------

P\_id|0|FT|3|P\_id|1|FT|10|P\_id|3|FT|5|P\_id|9|FT|15|P\_id|12|FT|20|

--------------------------------------------------------

Enter your choice

1.FCFS

2.SJF(premptive)

3.Priority(non-premptive)

4.Round Robin

5.quit3

Enter the number of processes

5

Enter the arrival times

0

1

3

9

12

Enter the burst times

3

5

2

5

5

Enter the priorities

3

1

3

2

1

P.Id Arrival Burst Prio Finish Turn Waiting

P1 0 3 3 3 3 0

P2 1 5 1 8 7 2

P3 3 2 3 10 7 5

P4 9 5 2 15 6 1

P5 12 5 1 20 8 3

Gantt Chart

-----------------------------------------------------

P\_id|0|FT|3|P\_id|1|FT|8|P\_id|3|FT|10|P\_id|9|FT|15|P\_id|12|FT|20|

--------------------------------------------------------

Enter your choice

1.FCFS

2.SJF(premptive)

3.Priority(non-premptive)

4.Round Robin

5.quit4

Enter the number of processes

5

Enter the arrival times

0

1

2

3

4

Enter the burst times

3

5

2

5

5

Enter the time slab

1

P.Id Arival Burst Finish Turn Waiting

P1 0 3 7 7 4

P2 1 5 16 15 10

P3 2 2 9 7 5

P4 3 5 19 16 11

P5 4 5 20 16 11

Gantt Chart

-----------------------------------------------------

P\_id|0|FT|7|P\_id|1|FT|16|P\_id|2|FT|9|P\_id|3|FT|19|P\_id|4|FT|20|

--------------------------------------------------------

Enter your choice

1.FCFS

2.SJF(premptive)

3.Priority(non-premptive)

4.Round Robin

5.quit5

poonam@ubuntu:~/Desktop$

\*\*\*--+--