|  |
| --- |
| **Lab 5:**   1. **Write a program to implement producer consumer problem** |

#include<stdio.h>

int main()

{

int buffer[5],bufsize,in,out,produce,consume,choice=0;

in= 0;

out = 0;

bufsize = 5;

while(choice!=3)

{

printf("\n1. Produce \t 2. Consume \t3. Exit");

printf("\nEnter your choice: ");

scanf("%d",&choice);

switch(choice)

{

case 1:

if((in+1)%bufsize==out)

printf("\nBuffer is Full and producer go to sleep");

else

{

printf("\nEnter the value: ");

scanf("%d", &produce);

buffer[in] = produce;

in = (in+1)%bufsize;

}

break;

case 2:

if(in == out)

printf("\nBuffer is Empty and consumer is goto sleep");

else

{

consume = buffer[out];

printf("\nThe consumed value is %d", consume);

out = (out+1)%bufsize;

}

break;

}

}

}

|  |
| --- |
| **b. Write a program to implement Round Robin algorithm** |

#include<stdio.h>

main()

{

int i,j,n,bu[10],wa[10],tat[10],t,ct[10],max;

float awt=0,att=0,temp=0;

printf("Enter the no of processes -- ");

scanf("%d",&n);

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

{

printf("\nEnter Burst Time for process %d -- ", i+1);

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

ct[i]=bu[i];

}

printf("\nEnter the size of time slice -- ");

scanf("%d",&t);

max=bu[0];

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

if(max<bu[i])

max=bu[i];

for(j=0;j<(max/t)+1;j++)

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

if(bu[i]!=0)

if(bu[i]<=t)

{

tat[i]=temp+bu[i];

temp=temp+bu[i];

bu[i]=0;

}

else

{

bu[i]=bu[i]-t;

temp=temp+t;

}

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

{

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

att+=tat[i];

awt+=wa[i];

}

printf("\nThe Average Turnaround time is -- %f",att/n);

printf("\nThe Average Waiting time is -- %f",awt/n);

printf("\n\tPROCESS\t BURST TIME \t WAITING TIME\tTURNAROUND TIME\n");

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

printf("\t%d \t %d \t\t %d \t\t %d \n",i+1,ct[i],wa[i],tat[i]);

}