Ex: NO 1 INSTALLATION OF WINDOWS XP OPERATING SYSTEM

Aim:

To install Windows XP Operating System.

Description

Windows XP is a versatile O.S. which took over various laggings in the earlier O.S. given by Microsoft. One major component which really improves the performance and working is the availability of various device drivers which really make device operation as simple as plug-n-play. If the minimum system requirement is not fulfilled then you will not be able to install this O.S. on to your machine, however, Windows 98 might serve the purpose. Windows XP is in huge demand globally, let us learn how to load it before making us to work on it.

System Requirements

System Requirements	**Minimum**	**Recommended**
Processor	233MHz	300 MHz or higher
Memory	64 MB RAM	128 MB RAM or higher
Video adapter and Monitor	Super VGA (800 x 600) or higher resolution	
Hard drive disk free space	1.5 GB or higher	
Drives	CD-ROM drive or DVD drive	
Input Devices	Keyborad, Microsoft Mouse	
Sound	Speakers. Sound card, Head Phones	

Installation Steps

The following step by step procedure will help you to install Windows XP. The installation procedure is shown with the figure appears on your screen after doing a step.

1) Insert the Windows XP CD into your computer and restart.

2) If prompted to start from the CD, press SPACEBAR. If you miss the prompt (it only appears for a few seconds), restart your computer to try again.

3) Windows XP Setup begins. During this portion of setup, your mouse will not work, so you must use the keyboard. On the Welcome to Setup page, press ENTER.

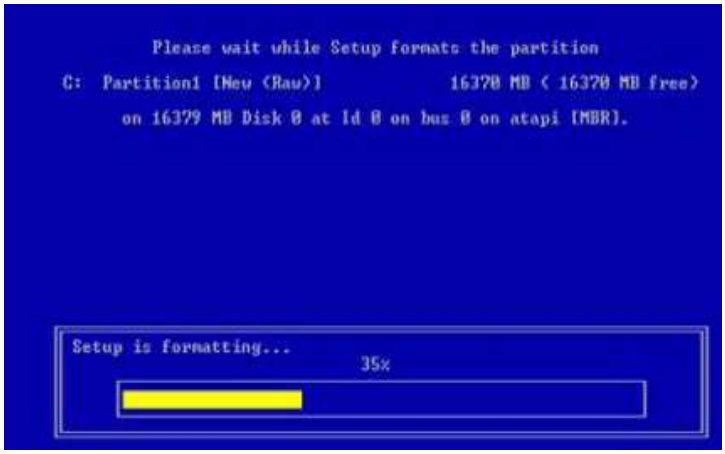
4) On the Windows XP Licensing Agreement page, read the licensing agreement. Press the PAGE DOWN key to scroll to the bottom of the agreement. Then press F8.

5) Next page enables you to select the hard disk drive on which Windows XP will be installed. Once you complete this step, all data on your hard disk drive will be removed and cannot be recovered. It is extremely important that you have a recent backup copy of your files before continuing. When you have a backup copy, press D, and then press L when prompted. This deletes your existing data.

6) Press ENTER to select Unpartitioned space, which appears by default.

7) Press ENTER again to select Format the partition using the NTFS file system, which appears by default.

8) Windows XP erases your hard disk drive using a process called formatting and then copies the setup files. You can leave your computer and return in 20 to 30 minutes



9) Windows XP restarts and then continues with the installation process. From this point forward, you can use your mouse. Eventually, the Regional and Language Options page appears. Click Next to accept the default settings. If you are multilingual or prefer a language other than English, you can change language settings after setup is complete.

10) On the Personalize Your Software page, type your name and your organization name. Some programs use this information to automatically fill in your name when required. Then, click Next.

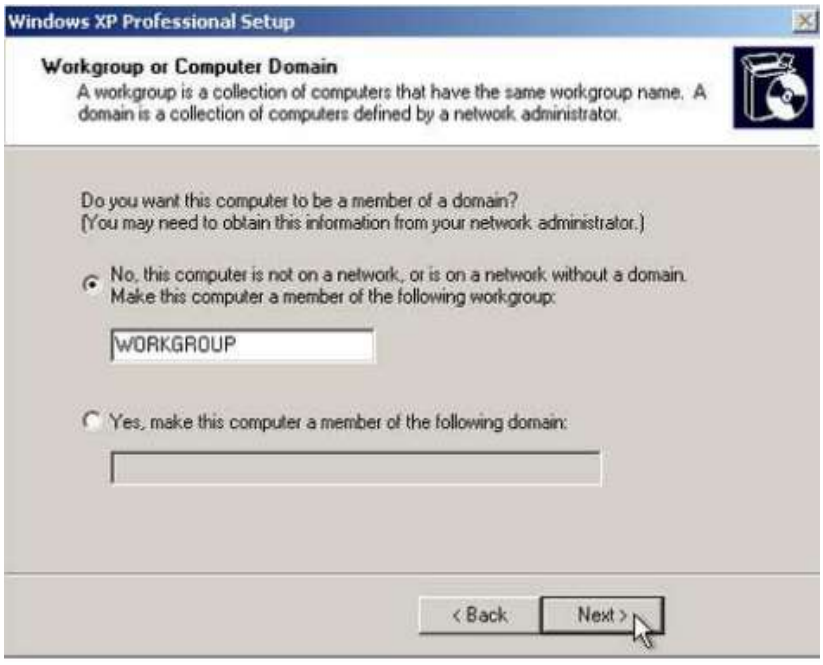
11) On the Your Product Key page, type your product key as it appears on your Windows XP CD case. The product key is unique for every Windows XP installation. Then, click Next.

12) On the Computer Name and Administrator Password page, in the Computer name box, type a name that uniquely identifies your computer in your house, such as FAMILYROOM or TOMS. You cannot use spaces or punctuation. If you connect your computer to a network, you will use this computer name to find shared files and printers. Type a strong password that you can remember in the Administrator password box, and then retype it in the Confirm password box. Write the password down and store it in a secure place. Click Next.

13) On the Date and Time Settings page, set your computer's clock. Then, click the Time Zone down arrow, and select your time zone. Click Next.

14) Windows XP will spend about a minute configuring your computer. On the Networking Settings page, click Next.

15) On the Workgroup or Computer Domain page, click Next.



16) Windows XP will spend 20 or 30 minutes configuring your computer and will automatically restart when finished. When the Display Settings dialog appears, click OK.

17) When the Monitor Settings dialog box appears, click OK.

18) The final stage of setup begins. On the Welcome to Microsoft Windows page, click Next.

19) On the Help protect your PC page, click Help protect my PC by turning on Automatic Updates now. Then, click Next

20) Windows XP will then check if you are connected to the Internet:

If you are connected to the Internet, select the choice that describes your network connection on the Will this computer connect to the Internet directly, or through a network? page. If you're not sure, accept the default selection, and click Next.

21) If you use dial-up Internet access, or if Windows XP cannot connect to the Internet, you can connect to the Internet after setup is complete. On the How will this computer connect to the Internet? page, click Skip.

22) Windows XP Setup displays the Ready to activate Windows? page. If you are connected to the Internet, click Yes, and then click Next. If you are not yet connected to the Internet, click No, click Next, and then skip to step 24. After setup is complete, Windows XP will automatically remind you to activate and register your copy of Windows XP.

23) On the Ready to register with Microsoft? page, click Yes, and then click Next

24) On the Collecting Registration Information page, complete the form. Then, click Next.

25) On the Who will use this computer? page, type the name of each person who will use the computer. You can use first names only, nicknames, or full names. Then click Next. To

add users after setup is complete or to specify a password to keep your account private, read Create and customize user accounts.

26) On the Thank you! page, click Finish.

27) Congratulations! Windows XP setup is complete. You can log on by clicking your name on the logon screen. If you've installed Windows XP on a new computer or new hard disk drive, you can now use the File and Settings Transfer Wizard to copy your important data to your computer or hard disk drive.

RESULT:

Thus the Windows Operating System is installed and Executed successfully.

PROGRAM:

A)Swapping values of two variables

```
echo -n ”Enter value for A:”
read a
echo -n “Enter value for B:”
read b
t=$a
a=$b
b=$t
echo “Values after Swapping”
echo “A Value is $a and B Value is $b”
```

B)Fahrenheit to centigrade conversion

```
echo -n “Enter Fahrenheit:”
read f
c = ‘expr \ ( $f – 32 \ ) \ * 5 / 9’
echo “Centigrade is :$c”
```

C)Biggest of 3 numbers

```
echo -n “Give value for A,B and C:”
read a b c
if [$a -gt $b -a $a -gt $c]
then
      echo “A is the Biggest number”
elif [ $b -gt $c]
then
     echo “B is the Biggest number”
else
    echo “C is the Biggest number”
fi
```

D)Grade determination

```
echo -n "Enter the mark:"
read mark
if [ $mark -gt 90 ]
then
    echo "S grade"
elif [ $mark -gt 80 ]
then
    echo "A grade"
elif [ $mark -gt 70 ]
then
    echo "B grade"
elif [ $mark -gt 60 ]
then
    echo "C grade"
elif [ $mark -gt 55 ]
then
    echo "D grade"
elif [ $mark -gt 90 ]
then
    echo "E grade"
else
    echo "U grade"
fi
```

E)Vowel or consonant

```
echo -n "Key in a lower case character:"
read choice
case $choice in
     a|e|i|o|u) echo "Its a Vowel";;
                *) echo "Its a consonant"
esac
```

F)Simple calculator

```
echo -n "Enter the two numbers:"
read a b
echo "1.Addition"
echo "2.Subtraction"
echo "3.Multiplication"
echo "4.Division"
echo -n "Enter the option:"
read option
case $option in
  1) c=`expr $a + $b`
```

```
        echo "$a+$b=$c";;
   2) c=`expr $a - $b`
        echo "$a-$b=$c";;
   3) c=`expr $a \* $b`
        echo "$a*$b=$c";;
   4) c=`expr $a / $b`
        echo "$a/$b=$c";;
        *) echo "Invalid option"
esac
```

G)Multiplication table

```
Clear
echo-n "Which multiplication table? : "
read n
for x in 1 2 3 4 5 6 7 8 9 10
do
    p = 'expr $x \* $n'
    echo -n "$n X $x = $p"
    Sleep 1
done
```

H)Number reverse

```
echo -n "Enter a number: "
read n
rd=0
while [ $n -gt 0 ]
do
    rem = ' expr $n % 10 '
    rd = ' expr $rd \* 10 + $rem '
    n = ' expr $n / 10 '
done
echo " Reversed number is $rd"
```

I)Prime Number

```
echo -n "Enter the number: "
read n
m = 'expr $n / 2"
until [ $i -gt $m ]
do
    q='expr $n % $i'
    if [ $q -eq 0 ]
    then
```

```
        echo "Not a prime number"
        exit
    fi
        i = 'expr $i +1'
done
echo "Prime number"
```

PROGRAM:

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/types.h>
int main()
{
pid_t pid;
int x=5;
pid=fork();
x++;
if(pid<0)
{
Printf("Process creation error");
exit(-1);
}
Else if(pid==0)
{
printf("Child process:"); printf("\nProcess
id is %d",getpid()); printf("\nvalue of x is
%d",x); printf("\nProcess id of parent is
%d\n",getppid());
}
```

PROGRAM:

```
#include<stdio.h>
#include<unistd.h> int
main()
{

int i,pid;
pid=fork();
if(pid==-1)
{

printf("fork failed");
exit(0);
}

else if(pid==0)

{
printf("\nChild process starts");
for(i=0;i<5;i++)
{
printf("\nChild process %d is called",i);

}
printf("\nChild process ends");

}

else

{

wait(0); printf("\nParent process
ends");
}

exit(0);

}
```

PROGRAM:

```
#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<sys/types.h>
#include<sys/wait.h>
main()
{

pid_t  pid;
switch(pid=fork())
{

case -1:

    perror("Fork failed");

    exit(-1);
case 0:
    printf("Child process\n");
execl("/bin/date",""date",NULL);
    exit(0);
default:
    wait(NULL);
printf("Child Terminated\n");
}
}
```

PROGRAM:

```
#include<stdio.h>
#include<unistd.h>
int main()
{
int pid,pid1,pid2;
pid=fork();
if(pid==-1)
{
printf("ERROR IN PROCESS CREATION\n");
exit(1);
}
If(pid!=0)
{
pid1=getpid();
printf("\n the parent process ID is %d\n",pid1);
}
Else
{
pid2=getpid();
printf("\n the child process ID is %d\n",pid2);
}
}
```

PROGRAM:

```
#include<stdio.h>
struct process
{
int pid;
int btime;
int wtime;
int ttime;
}p[10];
int main( )
{
int i,j,k,n,ttur,twat;
float awat,atur;
printf("Enter no.of.process:");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("Burst time for process P%d(in ms):",(i+1));
scanf("%d",&p[i].btime);
p[i].pid=i+1;
}
p[0].wtime=0;
for(i=0;i<n;i++)
{
p[i+1].wtime=p[i].wtime+p[i].btime;
p[i].ttime=p[i].wtime+p[i].btime;
}
ttur=twat=0;
for(i=0;i<n;i++)
{
ttur+=p[i].ttime;
twat+=p[i].wtime;
```

```
}
awat = (float)twat /n;
atur = (float)ttur / n;
printf("\nFCFS scheduling\n\n");
for(i=0;i<28;i++)
printf("-");
printf("\nProcess B-Time T-Time W-Time\n");
for(i=0;i<28;i++)
printf("-");
for(i=0;i<n;i++)
printf("\nP%d\t%4d\t%3d\t%2d",p[i].pid,p[i].btime,p[i].ttime,p[i].wtime);
printf("\n");
for(i=0;i<28;i++)
printf("-");
printf("Average waiting time:%5.2fms",awat);
printf("Average turn around time:%5.2fms",atur);
printf("\n\nGANTT CHART\n");
printf("-");
for(i=0;i<n;i++)
{
k=p[i].btime/2;
for(j=0;j,k;j++)
   printf("-");
printf("P%d",p[i].pid);
for(j=k+1;j<p[i].btime;j++)
   printf(" ");
printf("|");
}
printf("\n-");
for(i=0;i<(p[n-1].ttime+2*n);i++)
  printf("-");
```

```
printf(“\n”);
printf(“|”);
for(i=0;i<n;i++)
{
k=p[i].btime/2;
for(j=0;j,k;j++)
   printf(“-“);
printf(“P%d”,p[i].pid);
for(j=k+1;j<p[i].btime;j++)
   printf(“ “);
printf(“|”);
}
printf(“\n-“);
for(i=0;i<(p[n-1].ttime+2*n);i++)
  printf(“-“);
printf(“\n”);
printf(“0”);
for(i=0;i<n;i++)
{
for(j=0;j<p[i].btime;j++)
    printf(“ “);
printf(“%2d”,p[i].ttime);
}
}
```

PROGRAM:

```
#include<stdio.h>
struct process
{
int pid;
int btime;
int wtime;
int ttime;
}p[10],temp;
int main()
{
int i,j,k,n,ttur,twat;
float awat,atur;
printf("Enter no.of process: ");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("Burst time for process P%d(in ms): ",(i+1));
scanf("%d",&p[i].btime);
p[i].pid=i+1;
}
for(i=0;i<n-1;i++)
{
for(j=i+1;j<n;j++)
{
if(p[i].btime>p[j].btime)||(p[i].btime==p[j].btime&&p[i]>p[j].pid))
{
temp=p[i];
p[i]=p[j];
p[j]=temp;
}
}
}
p[0].wtime=0;
for(i=0;i<n;i++)
{
p[i+1].wtime=p[i].wtime+p[i].btime;
p[i].ttime=p[i].wtime+p[i].btime;
}
ttur=twat=0;
for(i=0;i<n;i++)
{
ttur+=p[i].ttime;
twat+=p[i].wtime;
}
awat=(float)twat/n;
atur=(float)ttur/n;
```

```
printf(“\n SJF Scheduling \n\n”);
for(i=0;i<28;i++)
printf(”-“);
printf(“\n Process B-time T-time W-time \n”);
printf(“\n0”);
for(i=0;i<n;i++)
{
for(j=0;j<p[i].btime;j++)
printf(” “);
printf(“%2d”,p[i].ttime);
}
}
```

PROGRAM:

```
#include<stdio.h>
struct process
{
int pid;
int pri;
int btime;
int ttime;
int wtime;
}p[10],temp;
int main()
{
int i,j,k,n,ttur,twat;
float awat,atur;
printf("enter no of process: ");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("Burst time for process p%d (in ms): ",(i+1));
scanf("%d",&p[i].btime);
printf("priority for process P%d(in ms): ",(i+1));
scanf("%d",&p[i].pri);
p[i].pid=i+1;
}
for(i=0;i<n-1;i++)
{
for(j=i+1;j<n;j++)
{
if((p[i].btime>p[j].btime)||(p[i].btime==p[j].btime&&p[i].pid>p[j].pid))
{
temp=p[i];
p[i]=p[j];
p[j]=temp;
}
}
}
p[0].wtime=0;
for(i=0;i<n;i++)
{
p[i+1].wtime=p[i].wtime+p[i].btime;
p[i].ttime=p[i].wtime+p[i].btime;
}
ttur=twat=0;
for(i=0;i<n;i++)
{
ttur+=p[i].ttime;
twat+=p[i].wtime;
}
```

```
awat=(float)twat/n;
atur=(float)ttur/n;
printf("\n PRIORITY scheduling\n\n");
for(i=0;i<28;i++)
printf("-");
printf("\n process B-time T-time W-time \n");
for(i=0;i<28;i++)
printf("-");
for(i=0;i<n;i++)
printf("\np%d\t%4d\t%3d\t%2d",p[i].pid,p[i].btime,p[i].ttime,p[i].wtime);
printf("\n");
for(i=0;i<28;i++)
printf("-");
for(i=0;i<n;i++)
{
k=p[i].btime/2;
for(j=0;j<k;j++)
printf("-");
printf("P %d",p[i].pid);
for(j=k+1;j<p[i].btime;j++)
printf(" ");
printf("|");
}
printf("\n-");
for(i=0;i<(p[n-1].ttime+2*n);i++)
printf("-");
printf("\n0");
for(i=0;i<n;i++)
{for(j=0;j<p[i].btime;j++)
printf(" ");
printf("%2d",p[i].ttime);
}
}
```

PROGRAM:

```
#include<stdio.h>
int main()
{
int i,x=-1,k[10],m=0,n,t,s=0;
int a[50],temp,b[50],p[10],bur[10],bur1[10];
int wat[10],tur[10],ttur=0,twat=0,j=0;
float awat,atur;
printf("enter no of process: ");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("Burst time for process p%d (in ms): ",(i+1));
scanf("%d",&bur[i]);
bur1[i]=bur[i];
}
printf("enter the time slice (in ms): ");
scanf("%d",&t);
for(i=0;i<n;i++)
{
b[i]=bur[i]/t;
if((bur[i]%t)!=0)
b[i]+=1;
m+=b[i];
}
printf("\n\t\t Round Robin Scheduling");
printf("\n GANTT chart \n");
for(i=0;i<n;i++)
printf("---------------");
printf("\n");
a[0]=0;
while(j<m)
{
if(x==n-1)
x=0;
else
x++;
if(bur[x]>=1)
{
bur[x]-=t;
a[j+1]=a[j]+t;
if(b[x]==1)
{
p[s]=x;
k[s]=a[j+1];
s++;
}
j++;
```

```
b[x]-=1;
printf("P%d|",x+1);
}
else if(bur[x]!=0)
{
a[j+1]=a[j]+bur[x];
bur[x]=0;
if(b[x]==1)
{
p[s]=x;
k[s]=a[j+1];
s++;
}
j++;
b[x]-=1;
printf("P%d|",x+1);
}
}
printf("\n");
for(i=0;i<m;i++);
printf("----------------");
printf("\n");
for(j=0;j<m;j++);
printf("%d\t",a[j]);
for(i=0;i<n;i++);
for(j=i+1;j<n;j++);
{
if(p[i]>p[j])
{
temp=p[i];
p[i]=p[j];
p[j]=temp;
temp=k[i];
k[i]=k[j];
k[j]=temp;
}
}
for(i=0;i<n;i++)
{
wat[i]=k[i]-bur1[i];
tur[i]=k[i];
}
for(i=0;i<n;i++);
{
ttur+=tur[i];
twat+=wat[i];
}
printf("\n\n");
```

```
for(i=0;i<30;i++)
printf("-");
printf("\n Process\tBurst\tTrnd\twait\n");
for(i=0;i<30;i++)
printf("-");
for(i=0;i<n;i++);
printf("\nP%-4d\t%4d\t%4d\t%4d",p[i]+1,bur1[i],tur[i],wat[i]);
printf("\n");
for(i=0;i<30;i++)
printf("-");
awat=(float)twat/n;
atur=(float)ttur/n;
printf("\n\n average waiting time:%2f ms",awat);
printf("\n\n average turn around time:%2f ms",atur);
}
```

PROGRAM:

```
#include<stdio.h>
#include<string.h>
#include<sys/types.h>
#include<stdlib.h>
#include<sys/wait.h>
#include<unistd.h>
#define msgsize 29
int main(void)
{
int ser[2],cli[2],p;
char inbuff[msgsize];
char*msg="Thank you";
system("clear");
pipe(ser);
pipe(cli);
printf("\nserver read id=%d,write id=%d",ser[0],ser[1]);
printf("\nclient read id=%d,write id=%d",cli[0],cli[1]);
p=fork();
if(p==0)
{
printf("\ni am in child process!");
close(cli[0]);
close(ser[1]);
write(cli[1],msg,msgsize);
printf("\nmessage written to pipe...!");
sleep(2);
read(ser[0],inbuff,msgsize);
printf("\necho message received from server");
printf("\n%s",inbuff);
}
else
{
close(cli[1]);
close(ser[0]);
printf("\nparent process");
read(cli[0],inbuff,msgsize);
printf("\nparent ended!");
}
}
```

PROGRAM:

```
#include<stdio.h>
#include<sys/ipc.h>
#include<string.h>
#include<sys/msg.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/wait.h>
#include<sys/types.h>
struct
{
long mtype;
char mtext[20];
}
send,recv;
int main()
{
int qid,pid,len;
qid=msgget((key_t)0x2000,IPC_CREAT|0666);
if(qid==-1)
{
perror("\n message failed");
exit(1);
}
send.mtype=1;
strcpy(send.mtext,"\nhello i am parent");
len=strlen(send.mtext);
pid=fork();
if(pid>0)
{
if(msgsnd(qid,&send,len,0)==-1)
{
perror("\n message sending failed");
exit(1);
}
printf("\nmessage has been posted");
sleep(2);
if(msgrcv(qid,&recv,100,2,0)==-1)
{
perror("\nmsgrev error:");
exit(1);
}
printf("\nmessage received from child -%s\n",recv.mtext);
}
else
{
send.mtype=2;
```

```
strcpy(send.mtext,"\n hi i am child");
len=strlen(send.mtext);
if(msgrcv(qid,&recv,100,1,0)==-1)
{
perror("\nchild message received failed");
exit(1);
}
if(msgsnd(qid,&send,len,0)==-1)
{
perror("\nchild message send failed");
}
printf("\n received from parent-%s",recv.mtext);
}
}
```

PROGRAM:

```
#include<stdio.h>
#include<sys/ipc.h>
#include<string.h>
#include<sys/types.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/shm.h>
#define SEGSIZE 100
int main(int argc,char* argv[])
{
int shmid,cntr;
key_t key;
char* segptr;
char buff[]="pooda...";
key=ftok(".",'s');
if((shmid==shmget(key,SEGSIZE,IPC_CREAT|IPC_EXCL|0666))==-1)
{
if((shmid=shmget(key,SEGSIZE,0))==-1)
{
perror("shmget");
exit(1);
}
}
else
{
printf("Creating a new shared memory seg\n");
printf("SHMID:%d",shmid);
}
system("ipcs -m");
if((segptr=(char*)shmat(shmid,0,0))==(char*)-1)
{
perror("shmat");
exit(1);
}
printf("Writing data to shared memory..\n");
strcpy(segptr,buff);
printf("DONE\n");
printf("Reading data from shared memory...\n");
printf("DATA:-%s\n",segptr);
printf("DONE\n");
printf("Reading data from shared memory segment...\n");
if(shmctl(shmid,IPC_RMID,0)==-1)
printf("Can't Remove shared memory segment...\n");
else
printf("Removed Successfully");
}
```

PROGRAM:

```
#include<stdio.h>
#include<pthread.h>
#include<semaphore.h>
#include<unistd.h>
sem_t mutex;
void * thread(void*arg)
{
sem_wait(&mutex);
printf("\nEntered..\n");
sleep(4);
printf("\nJust exiting...\n");
sem_post(&mutex);
}
int main()
{
sem_init(&mutex,0,1);
pthread_t t1,t2;
pthread_create(&t1,NULL,thread,NULL);
sleep(2);
pthread_create(&t2,NULL,thread,NULL);
pthread_join(t1,NULL);
pthread_join(t2,NULL);
sem_destroy(&mutex);
return 0;
}
```

PROGRAM:

```
#include<stdio.h>
main()
{
int r[1][10],av[1][10];
int all[10][10],max[10][10],ne[10][10],w[10],safe[10];
int i=0,j=0,k=0,np=0,nr=0,count=0,cnt=0;
printf(“enter the number of processes in a system”);
scanf(“%d”,&np);
printf(“enter the number of resourses in a system”);
scanf(“%d”,&nr);
for(i=1;i<=nr;i++)
printf(“enter the number of instances of resource R%d,i”);
scanf(“%d”,&r[0][i]);
av[0][i]=r[0][i];
}
for(i=1;i<=np;i++){
for(j=1;j<=nr;j++){
all[i][j]=ne[i][j]=max[i][j]=w[i]=0;
printf(“enter the allocation matrix”;
}
}
for(i=1;i<=np;i++){
for(j=1;j<=nr;j++){
scanf(“%d”,&all[i][j]);
av[0][i]=av[0][j]-all[i][j];
}
}
printf(“enter the maximum matrix”);
for(i=1;i<=np;i++) {
for(j=1;j<=nr;j++) {
scanf(“%d”&max[i][j]);
}
}
for(i=1;i<=np;i++) {
for(j=1;j<=nr;j++)
{
ne[i][j]=max[i][j]-all[i][j];
}
}
for(i=1;i<=np;i++)
{
printf(“process P %d”,i);
for(j=1;j<=nr;j++)
{
printf(“\n allocated %d\t”,all[i][j]’’);
printf(“\n maximum %d\t”,max[i][j]’’);
printf(“\n need %d\t”,ne[i][j]’’);
```

```
}
printf(“\n_____\n’’;
}
printf(“\n availability’);
for(i=1;i<=nr;i++)
printf(“\nR%d%d\t”,i,av[0][i]);
printf(“\n_____\n’’);
printf(“\nsafe sequence’’);
for (count=1,count<=np;count++)
{
for(i=1;i<=np;i++)
{
cnt=0;
for(j=1;j<=nr;j++) {
if(ne[i][j]<=av[0][j]&&w[i]==0)
cnt++;
}
if(cnt==nr)
{
k++;
safe[k]=i;
for(l=1;l<=nr;l++)
av[0][1]=av[0][1]+all[i][l];
printf(“\nP %d)”,safe[k]);
printf(“\t Availability’’);
for(l=1;l<=nr;l++)
printf(“R %d%d\t”,l ,av[0][1]);
w[i]=1;
}
}
}
}
```

PROGRAM:

```
#include<stdio.h>
int max[100][100];
int aloc[100][100];
int need[100][100];
int avail[100];
int n,r;
void input( );
void show( );
void cal( );
main( )
{
int i,j;
printf("Deadlock Detection Algo\n");
input( );
show( );
cal( );
}
void input( )
{
int i,j;
printf("Enter the no.of.Processes\t");
scanf("%d",&n);
printf("Enter the no.of.resource instances\t");
scanf("%d",&r);
printf("Enter the Max Matrix\n");
for(i=0;i<n;i++)
for(j=0;j<r;j++)
scanf("%d",&max[i][j]);
printf("Enter the Allocation Matrix\n");
for(i=0;i<n;i++)
```

```
for(j=0;j<r;j++)
scanf(“%d”,&alloc[i][j]);
printf(“Enter the available Resources\n”);
for(j=0;j<r;j++)
scanf(“%d”,&avail[j]);
}
void show( )
{
int i,j;
printf(“Process\tAllocation\tMax\tAvailable\t”);
for(i=0;i<n;i++)
{
printf(“\nP%d\t”,i+1);
For(j=0;j<r;j++)
{
printf(“%d”,alloc[i][j]);
}
printf(“\t”);
for(j=0;j<r;j++)
{
printf(“%d”,max[i][j]);
}
Printf(“\t”);
If(i==0)
{
for(j=0;j<r;j++)
Printf(“%d”,avail[j]);
}
}
}
void cal()
```

```
{
int finish[100],temp,need[100][100],flag=1,k,cl=0;
int dead[100];
int safe[100];
int i,j;
for(i=0;i<n;i++)
{
finish[i]=0;
}
for(i=0;i<n;i++)
{
for(j=0;j<r;j++){
need[i][j]=max[i][j]-alloc[i][j];
}
}
while(flag)
{
flag=0;
for(i=0;i<n;i++)
{
int c=0;
for(j=0;j<r;j++)
{
if(finish[i]==0)&&(need[i][j]<=avail[j])
{
c++;
if(c==r)
{
for(k=0;k<r;k++)
{
avail[k]+=alloc[i][j];
finish[i]=1;
```

```
flag=1;
}
if(finish[i]==1)
{
i=n;
} } } } } }
j=0;
flag=0;
for(i=0;i<n;i++)
{
if(finish[i]==0)
{
dead[j]=I;
j++;
flag=1;
}
}
if(flag==1)
{
printf("\n\nSystem is in Deadlock and the Deadlock process ara\n");
for(i=0;i<n;i++)
{
printf("P%d\t",dead[i]);
}
}
else
{
printf("\nNo Deadlock Occur");
}
}
```

PROGRAM:

```
#include<stdio.h>
#include<string.h>
#include<pthread.h>
#include<stdlib.h>
#include<unistd.h>
pthread_t tid[2];
void * doSomeThing(void * arg)
{
unsigned long i=0;
pthread_t id=pthread_self();
if(pthread_equal(id,tid[0]))
{
printf("\nFirst Thread Processing\n");
}
else
{
printf("\nSecond Thread Processing\n");
}
for(i=0;i<(0xFFFFFFFF);i++)
return NULL;
}
int main(void)
{
int i=0;
int err;
while(i<2)
{
err=pthread_create(&(tid[i]),NULL,&doSomeThing,NULL0;
if(err!=0)
printf("\ncan't create thread:[%s]",strerror(err));
```

```
else
printf(“\nThread created successfully\n”);
i++;
}
sleep(5);
return 0;
}
```

PROGRAM:

```
#include<stdio.h>
#include<math.h>
main( )
{
int size,m,n,pgno,pagetable[3]={5,6,7},I,j,frameno;
double m1;
int ra=0;ofs;
printf("Enter process size(in KB of max 12KB):");
scanf("%d",&size);
m1=size/4;
n=ceil(m1);
printf("Total No.of.pages:%d",n);
printf("\nEnter relative address (in hexa)\n");
scanf("%d",&ra);
pgno=ra / 1000;
ofs=ra % 1000;
printf("page no=%d\n",pgno);
printf("page table");
for(i=0;i<n;i++)
printf("\n%d[%d]",i,pagetable[i]);
frameno=pagetable[pgno];
printf("\nPhysical address:%d%d",frameno,ofs);
}
```

PROGRAM:

```
#include<stdio.h>
#define max 25
void main()
{
int frag[max],b[max],f[max],i,j,nb,nf,temp;
static int bf[max],ff[max];
printf("\n\tMemory Management Scheme - First Fit");
printf("\nEnter the number of blocks:");
scanf("%d",&nb);
printf("Enter the number of files:");
scanf("%d",&nf);
printf("\nEnter the size of the blocks:-\n");
for(i=1;i<=nb;i++)
{
printf("Block %d:",i);
scanf("%d",&b[i]);
}
printf("Enter the size of the files :-\n");
for(i=1;i<=nf;i++)
{
printf("File %d:",i);
scanf("%d",&f[i]);
}
for(i=1;i<=nf;i++)
{
for(j=1;j<=nb;j++)
{
if(bf[j]!=1)
{
temp=b[j]-f[i];
if(temp>=0)
{
ff[i]=j;
break;
}
}
}
frag[i]=temp;
bf[ff[i]]=1;
}
printf("\nFile_no:\tFile_size :\tBlock_no:\tBlock_size:\tFragement");
for(i=1;i<=nf;i++)
printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],ff[i],b[ff[i]],frag[i]);
}
```

PROGRAM:

```
#include<stdio.h>
#define max 25
int main()
{
int frag[max],b[max],f[max],i,j,nb,nf,temp,highest;
static int bf[max],ff[max];
printf("\n\tMemory Management Scheme-First Fit");
printf("\nEnter the number of blocks:");
scanf("%d",&nb);
printf("\nEnter the number of files:");
scanf("%d",&nf);
printf("\nEnter the size of the blocks:\n");
for(i=1;i<=nb;i++)
{
printf("Block %d",i);
scanf("%d",&b[i]);
}
printf("Enter the size of the files:\n");
for(i=1;i<=nf;i++)
{
printf("File %d:",i);
scanf("%d",&f[i]);
}
for(i=1;i<+nf;i++)
{
for(j=1;j<=nb;j++)
{
if(bf[j]!=1)
{
temp=b[j]-f[i];
if(temp>=0)
{
ff[i]=j;
break;
}
}
}
frag[i]=highest;
bf[ff[i]]=1;
highest=0;
}
printf("\nFile_no:\tFile_size :\tBlock_no:\tBlock_size:\tFragement");
for(i=1;i<=nf;i++)
printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],ff[i],b[ff[i]],frag[i]);
}
```

PROGRAM:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define max 25
void main()
{
int frag[max],b[max],f[max],i,j,nb,nf,temp,lowest=10000;
static int bf[max],ff[max];
printf("\nEnter the number of blocks:");
scanf("%d",&nb);
printf("Enter the number of files:");
scanf("%d",&nf);
printf("\nEnter the size of the blocks:-\n");
for(i=1;i<=nb;i++)
{
printf("Block %d:",i);
scanf("%d",&b[i]);
}
printf("Enter the size of the files :-\n");
for(i=1;i<=nf;i++)
{
printf("File %d:",i);
scanf("%d",&f[i]);
}
for(i=1;i<=nf;i++)
{
for(j=1;j<=nb;j++)
{
if(bf[j]!=1){
temp=b[j]-f[i];
if(temp>=0)
if(lowest>temp)
{
ff[i]=j;
lowest=temp;
}
}
}
frag[i]=lowest;
bf[ff[i]]=1;
lowest=10000;
}
printf("\nFile No\tFile Size \tBlock No\tBlock Size\tFragment");
for(i=1;i<=nf && ff[i]!=0;i++)
printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],ff[i],b[ff[i]],frag[i]);
}
```

PROGRAM:

```
#include <stdio.h>
void main()
{
int i,j,l,rs[50],frame[10],nf,k,avail,count=0;
printf("Enter length of ref. string : ");
scanf("%d", &l);
printf("Enter reference string :\n");
for(i=1; i<=l; i++)
scanf("%d", &rs[i]);
printf("Enter number of frames : ");
scanf("%d", &nf);
for(i=0; i<nf; i++)
frame[i] = -1;
j = 0;
printf("\nRef. str Page frames");
for(i=1; i<=l; i++)
{
printf("\n%4d\t", rs[i]);
avail = 0;
for(k=0; k<nf; k++)
if(frame[k] == rs[i])
avail = 1;
if(avail == 0)
{
frame[j] = rs[i];
j = (j+1) % nf;
count++;
for(k=0; k<nf; k++)
printf("%4d", frame[k]);
}
}
printf("\n\nTotal no. of page faults : %d\n",count);
}
```

PROGRAM:

```
#include <stdio.h>
int arrmin(int[], int);
int main()
{
int i,j,len,rs[50],frame[10],nf,k,avail,count=0;
int access[10], freq=0, dm;
printf("Length of Reference string : ");
scanf("%d", &len);
printf("Enter reference string :\n");
for(i=1; i<=len; i++)
scanf("%d", &rs[i]);
printf("Enter no. of frames : ");
scanf("%d", &nf);
for(i=0; i<nf; i++)
frame[i] = -1;
j = 0;
printf("\nRef. str Page frames");
for(i=1; i<=len; i++)
{
printf("\n%4d\t", rs[i]);
avail = 0;
for(k=0; k<nf; k++)
{
if(frame[k] == rs[i])
{
avail = 1;
access[k] = ++freq;
break;
}
}
if(avail == 0)
{
dm = 0;
for(k=0; k<nf; k++)
{
if(frame[k] == -1)
dm = 1;
break;
}
if(dm == 1)
{
frame[k] = rs[i];
access[k] = ++freq;
count++;
}
else
```

```
{
j = arrmin(access, nf);
frame[j] = rs[i];
access[j] = ++freq;
count++;
}
for(k=0; k<nf; k++)
printf("%4d", frame[k]);
}
}
printf("\n\nTotal no. of page faults : %d\n", count);
}
int arrmin(int a[], int n)
{
int i, min = a[0];
for(i=1; i<n; i++)
if (min > a[i])
min = a[i];
for(i=0; i<n; i++)
if (min == a[i])
return i;
}
```

PROGRAM:

```
#include<stdio.h>
int i,j,nof,nor,flag=0,ref[50],frm[50],pf=0,victim=-1;
int recent[10],optcal[50],count=0;
int optvictim();
void main()
{
printf("\n OPTIMAL PAGE REPLACEMENT ALGORITHN");
printf("\n.....");
printf("\nEnter the no.of frames");
scanf("%d",&nof);
printf("Enter the no.of reference string");
scanf("%d",&nor);
printf("Enter the reference string");
for(i=0;i<nor;i++)
scanf("%d",&ref[i]);
printf("\n OPTIMAL PAGE REPLACEMENT ALGORITHM");
printf("\n.....");
printf("\nThe given string");
printf("\n.....\n");
for(i=0;i<nor;i++)
printf("%4d",ref[i]);
for(i=0;i<nof;i++)
{
frm[i]=-1;
optcal[i]=0;
}
for(i=0;i<10;i++)
recent[i]=0;
printf("\n");
for(i=0;i<nor;i++)
{
flag=0;
printf("\n\tref no %d ->\t",ref[i]);
for(j=0;j<nof;j++)
{
if(frm[j]==ref[i])
{
flag=1;
break;
}
}
if(flag==0)
{
count++;
if(count<=nof)
victim++;
else
```

```
victim=optvictim(i);
pf++;
frm[victim]=ref[i];
for(j=0;j<nof;j++)
printf("%4d",frm[j]);
}
}
printf("\n Number of page faults: %d",pf);
}
int optvictim(int index)
{
int i,j,temp,notfound;
for(i=0;i<nof;i++)
{
notfound=1;
for(j=index;j<nor;j++)
if(frm[i]==ref[j])
{
notfound=0;
optcal[i]=j;
break;
}
if(notfound==1)
return i;
}
temp=optcal[0];
for(i=1;i<nof;i++)
if(temp<optcal[i])
temp=optcal[i];
for(i=0;i<nof;i++)
if(frm[temp]==frm[i])
return i;
return 0;
}
```

PROGRAM:

```
#include<stdio.h>
#include<string.h>
int main()
{
int master,s[20],i,j,k;
char f[20][20][20],d[20][20];
printf("Enter number of directories: ");
scanf("%d",&master);
printf("Enter names of directories : ");
for(i=0;i<master;i++)
scanf("%s",d[i]);
for(i=0;i<master;i++)
{
printf("Enter number of files in directory %d :",i+1);
scanf("%d",&s[i]);
}
for(i=0;i<master;i++)
{
printf("Enter names of files in directory %d :",i+1);
for(j=0;j<s[i];j++)
{
scanf("%s",f[i][j]);
for(k=0;k<j;k++)
{
if(strcmp(f[i][j],f[i][k])==0)
{
printf("File Exists...");
printf("Enter new file name");
scanf("%s",f[i][j]);
}
}
}
printf("\n");
}
printf("Directory\t Size\t File name\n");
printf(" *************************\n");
for(i=0;i<master;i++)
{
printf("%s\t\t%2d\t",d[i],s[i]);
for(j=0;j<s[i];j++)
printf("%s\t\t",f[i][j]);
printf("\n");
}
printf("\t\n");
}
```

PROGRAM:

```
#include<stdio.h>
struct st
{
char dname[20],sdname[20][20],fname[20][20][20];
int ds,sds[20];
}
dir[20];
int main()
{
int i,j,k,n;
printf("Enter no.of.master file directories:");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("Enter Name of directory %d",i+1);
scanf("%s",dir[i].dname);
printf("Enter number of user file directories :");
scanf("%d",&dir[i].ds);
for(j=0;j<dir[i].ds;j++)
{
printf("Enter user file directory name and size");
scanf("%s",dir[i].sdname[j]);
scanf("%d",&dir[i].sds[j]);
for(k=0;k<dir[i].sds[j];k++)
{
printf("Enter file name :");
scanf("%s",dir[i].fname[j][k]);
}
}
}
printf("\n Master dir name\tsize\t sub dir name\t size\t files\n");
printf("\n*****************************************\n");
for(i=0;i<n;i++)
{
printf("%s\t\t%d",dir[i].dname,dir[i].ds);
for(j=0;j<dir[i].ds;j++)
{
printf("\t%s\t\t%d\t",dir[i].sdname[j],dir[i].sds[j]);
for(k=0;k<dir[i].sds[j];k++)
printf("%s\t",dir[i].fname[j][k]);
printf("\n\t\t");
}
printf("\n");
}
}
```

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
main()
{
 int n,i,j,b[20],sb[20],t[20],x,c[20][20];
 clrscr();
 printf("Enter no.of files:");
 scanf("%d",&n);
 for(i=0;i<n;i++)
 {
printf("Enter no. of blocks occupied by file%d",i+1);
scanf("%d",&b[i]);
printf("Enter the starting block of file%d",i+1);
scanf("%d",&sb[i]);
t[i]=sb[i];
for(j=0;j<b[i];j++)
c[i][j]=sb[i]++;
}
 printf("Filename\tStart block\tlength\n");
 for(i=0;i<n;i++)
printf("%d\t %d \t%d\n",i+1,t[i],b[i]);
 printf("Enter file name:");
 scanf("%d",&x);
 printf("File name is:%d",x);
 printf("length is:%d",b[x-1]);
 printf("blocks occupied:");
 for(i=0;i<b[x-1];i++)
printf("%4d",c[x-1][i]);
 getch();
}
```

PROGRAM

```
#include<stdio.h>
#include<conio.h>
main()
{
 int n,m[20],i,j,sb[20],s[20],b[20][20],x;
 clrscr();
 printf("Enter no. of files:");
 scanf("%d",&n);
 for(i=0;i<n;i++)
 { printf("Enter starting block and size of file%d:",i+1);
 scanf("%d%d",&sb[i],&s[i]);
 printf("Enter blocks occupied by file%d:",i+1);
 scanf("%d",&m[i]);
 printf("enter blocks of file%d:",i+1);
 for(j=0;j<m[i];j++)
 scanf("%d",&b[i][j]);
 } printf("\nFile\t index\tlength\n");
 for(i=0;i<n;i++)
 {
 printf("%d\t%d\t%d\n",i+1,sb[i],m[i]);
 }printf("\nEnter file name:");
 scanf("%d",&x);
 printf("file name is:%d\n",x);
 i=x-1;
 printf("Index is:%d",sb[i]);
 printf("Block occupied are:");
 for(j=0;j<m[i];j++)
 printf("%3d",b[i][j]);
 getch();
}
```

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
struct file
{
 char fname[10];
 int start,size,block[10];
}f[10];
main()
{
 int i,j,n;
 clrscr();
 printf("Enter no. of files:");
 scanf("%d",&n);
 for(i=0;i<n;i++)
 {
 printf("Enter file name:");
 scanf("%s",&f[i].fname);
 printf("Enter starting block:");
 scanf("%d",&f[i].start);
 f[i].block[0]=f[i].start;
 printf("Enter no.of blocks:");
 scanf("%d",&f[i].size);
 printf("Enter block numbers:");
 for(j=1;j<=f[i].size;j++)
 {
 scanf("%d",&f[i].block[j]);
 }
 }
 printf("File\tstart\tsize\tblock\n");
 for(i=0;i<n;i++)
 {
 printf("%s\t%d\t%d\t",f[i].fname,f[i].start,f[i].size);
 for(j=1;j<=f[i].size-1;j++)
  printf("%d--->",f[i].block[j]);
 printf("%d",f[i].block[j]);
 printf("\n");
 }
 getch();
}
```

PROGRAM:

```
#include<stdio.h>
int main()
{
int queue[20],n,head,i,j,k,seek=0,max,diff;
float avg;
printf("Enter the max range of disk\n");
scanf("%d",&max);
printf("Enter the size of queue request\n");
scanf("%d",&n);
printf("Enter the queue of disk positions to be read\n");
for(i=1;i<=n;i++)
scanf("%d",&queue[i]);
printf("Enter the initial head position\n");
scanf("%d",&head);
queue[0]=head;
for(j=0;j<=n-1;j++)
{
diff=abs(queue[j+1]-queue[j]);
seek+=diff;
printf("Disk head moves from %d to %d with seek %d\n”, queue[j],queue[j+1],diff);
}
printf("Total seek time is %d\n",seek);
avg=seek/(float)n;
printf("Average seek time is %f\n",avg);
return 0;
}
```

PROGRAM:

```
#include<stdio.h>
int main()
{
int queue[20],n,head,i,j,k,seek=0,max,diff,temp,queue1[20],queue2[20],
temp1=0,temp2=0;
float avg;
printf("Enter the max range of disk\n");
scanf("%d",&max);
printf("Enter the initial head position\n");
scanf("%d",&head);
printf("Enter the size of queue request\n");
scanf("%d",&n);
printf("Enter the queue of disk positions to be read\n");
for(i=1;i<=n;i++)
{
scanf("%d",&temp);
if(temp>=head)
{
queue1[temp1]=temp;
temp1++;
}
else
{
queue2[temp2]=temp;
temp2++;
}
}
for(i=0;i<temp1-1;i++)
{
for(j=i+1;j<temp1;j++)
{
if(queue1[i]>queue1[j])
{
temp=queue1[i];
queue1[i]=queue1[j];
queue1[j]=temp;
}
}
}
for(i=0;i<temp2-1;i++)
{
for(j=i+1;j<temp2;j++)
{
if(queue2[i]<queue2[j])
{
temp=queue2[i];
queue2[i]=queue2[j];
```

```
queue2[j]=temp;
}
}
}
for(i=1,j=0;j<temp1;i++,j++)
queue[i]=queue1[j];
queue[i]=max;
for(i=temp1+2,j=0;j<temp2;i++,j++)
queue[i]=queue2[j];
queue[i]=0;
queue[0]=head;
for(j=0;j<=n+1;j++)
{
diff=abs(queue[j+1]-queue[j]);
seek+=diff;
printf("Disk head moves from %d to %d with seek %d\n", queue[j],queue[j+1],diff);
}
printf("Total seek time is %d\n",seek);
avg=seek/(float)n;
printf("Average seek time is %f\n",avg);
return 0;
}
```

PROGRAM:

```
#include<stdio.h>
int main()
{
int queue[20],n,head,i,j,k,seek=0,max,diff,temp,queue1[20],queue2[20],
temp1=0,temp2=0;
float avg;
printf("Enter the max range of disk\n");
scanf("%d",&max);
printf("Enter the initial head position\n");
scanf("%d",&head);
printf("Enter the size of queue request\n");
scanf("%d",&n);
printf("Enter the queue of disk positions to be read\n");
for(i=1;i<=n;i++)
{
scanf("%d",&temp);
if(temp>=head)
{
queue1[temp1]=temp;
temp1++;
}
else
{
queue2[temp2]=temp;
temp2++;
}
}
for(i=0;i<temp1-1;i++)
{
for(j=i+1;j<temp1;j++)
{
if(queue1[i]>queue1[j])
{
temp=queue1[i];
queue1[i]=queue1[j];
queue1[j]=temp;
}
}
}
for(i=0;i<temp2-1;i++)
{
for(j=i+1;j<temp2;j++)
{
if(queue2[i]>queue2[j])
{
temp=queue2[i];
queue2[i]=queue2[j];
```

```
}
}
}
for(i=1,j=0;j<temp1;i++,j++)
queue[i]=queue1[j];
queue[i]=max;
queue[i+1]=0;
for(i=temp1+3,j=0;j<temp2;i++,j++)
queue[i]=queue2[j];
queue[0]=head;
for(j=0;j<=n+1;j++)
{
diff=abs(queue[j+1]-queue[j]);
seek+=diff;
printf("Disk head moves from %d to %d with seek %d\n”, queue[j],queue[j+1],diff);}
printf("Total seek time is %d\n",seek);
avg=seek/(float)n;
printf("Average seek time is %f\n",avg);
return 0;
}
```

Ex: No 16 **INSTALL LINUX OPERATING SYSTEM USING VMWARE.**

Aim:

To install any guest operating system like linux using VMware

Description:

Installing a Linux operating System using a VMware consists of two steps.

Part 1: Prepare a Computer for Virtualization

Part 2: Create a Virtual Machine

Scenario:

Personal computing power and resources have increased tremendously over the last 5 to 10 years. One of the benefits of having access to multicore processors and large amounts of RAM memory is the ability to use virtualization on a personal computer. With virtualization, a user can run multiple virtual computers on one physical computer or server. Virtual computers that run within a physical computer system are called virtual machines. Today entire computer networks are being implemented where all of the end user computer stations are actually virtual machines run off of a centralized server. Anyone with a modern computer and operating system has the ability to run virtual machines from the desktop.

Part 1: Prepare a Computer for Virtualization

In Part 1, you will download and install virtualization software, and acquire a bootable image of a Linux

distribution.

Step 1: Download and install the free VMware player.

There are two excellent virtualization programs that you can download and install for free, VMware Player and VirtualBox. In this lab, you will use the VMware player.

a. Go to http://vmware.com, hover the cursor over **Downloads**, and search for **Free Product Downloads**.

b. Under **Free Product Downloads**, click **Player**.

The VMware Player has 32-bit and 64-bit versions for Windows and Linux. To download the software, you

must register a free user account with VMware.

Note: The Linux version of VMware Player might work on Mac OS X; if not, http://virtualbox.org has a free

version of its VirtualBox software that will run on Mac OS X.

c. When you have downloaded the VMware Player installation file, run the installer and accept the default

installation settings.

Step 2: Download a bootable image of Linux.

You need an operating system to install on your virtual machine's virtual hardware. Linux is a suitable choice for an operating system, because most of the distributions are free to download and run. It also allows you to

explore an operating system that might be unfamiliar to you.

a. To download Linux, you first must select a distribution such as Mint, Ubuntu, Fedora, Suse, Debian, or

CentOS. (There are many others to choose from.) In this lab, the instructions follow an installation of Linux Mint.

b. Go to http://linuxmint.com, hover over the **Download** button, and click **Linux Mint 16** (or current version number).

c. Scroll down the page until you see the version of the Mint code name, **Cinnamon** (or the current code

name). Choose either the 32-bit or 64-bit version, depending on your current operating system platform,

and click the link.

d. A new web page will appear. Select a download mirror site from which to download the operating system. Click a mirror site to activate the Linux file download. When prompted by the browser, choose to save the

Linux .iso file to a local directory.

e. When the file has finished downloading, you should have a Linux Mint .iso bootable image.

Part 2: Create a Virtual Machine

In Part 2, using the VMware Player, you will create a virtual machine and customize its virtual hardware.

Then, using the Linux Mint .iso file that you downloaded in Part 1, you will install the Linux Mint operating

system on the virtual machine.

Step 1: Create a virtual computer and customize the virtual hardware.

a. Open **VMware Player**, and click **Create a New Virtual Machine**.

b. A new window will appear. Select **I will install the operating system later. The virtual machine will be**

created with a blank hard disk option. Click **Next**.

c. A new window will appear. Select **Linux** as the guest operating system. Under **version**, you may notice

that Mint is not listed. In this case, select an alternate Linux distribution, one that is closely related to Mint

(like Ubuntu). Lastly, select either the 32-bit or 64-bit version and click **Next**.

d. A new window will appear. Select a name and storage location for the virtual machine.

e. A new window will appear. Select the maximum size of the virtual hard drive storage. You can also decide

whether or not to store the virtual hard drive in a single file or in multiple files.

f. When a new window appears, click **Finish** to finish creating the virtual machine hardware, or click

Customize Hardware to customize the virtual hardware. For example, you can specify the amount of

RAM, how many CPU cores you want, and add additional drives and peripheral components (see video

tutorial).

g. When you have customized and completed the process, you are now ready to run your virtual machine

and install an operating system.

Step 2: Install the Linux operating system on the virtual computer.

To install from an .iso bootable image file, complete the following steps:

a. In VMware Player, click **Edit virtual machine settings**. If you have multiple virtual machines created, you

must first select which one you intend to edit.

b. In the pop-up window, under the **Hardware** tab, select **CD/DVD (SATA)**, and on the right side (under

Connections), select the **Use ISO image file** option, and click **Browse** to search through your file

system for the Linux Mint .iso image file downloaded in Part 1.

Selecting the Linux .iso file causes it to be automatically mounted to the CD/DVD drive when the virtual

machine boots, which, in turn, causes the system to boot and run the Linux installation image.

c. Select **Network Adapter**, and click **OK**.

The network adapter is currently set to NAT mode, which allows your virtual machine to access the

network through the host computer system by translating all Internet requests through the VMware player,

much like a gateway router. This gives your virtual machine a private network address separate from your

home network.

(Optional) To have your virtual machine on the same network as the computer hosting the virtual

machine, select the **Bridged** option and click **Configure Adapters** to identify to which physical network

adapter the virtual machine should bridge.

Note: This is especially important on laptops that have both wireless and wired network adapters that can

connect to the network.

d. When you are finished click **OK**.

e. Click **Play virtual machine** to launch your virtual machine and boot to Linux.

When the boot process has completed you should be presented with a Linux Mint desktop.

f. (Optional) To permanently install Linux Mint on the virtual machine hard disk drive, on the desktop,

double-click the **Install Linux Mint disk** icon and follow the installation procedure (see video).

During the installation, you will be presented with a warning message that installing to the drive will erase

everything on the disk drive; this refers to the virtual disk drive, not the host computer physical disk drive.

g. When you have finished the installation procedure, you should have a Linux Mint virtual machine that you

can run in a window alongside your host computer system.

Result:

Thus the guest operating system like linux using VMware is successfully installed.