**Experiment 4)**

**MFT Program**

4) Write a Program to implement Multiprogramming with a fixed number of tasks (MFT)

**Aim:** To implement a program for Multiprogramming with a fixed number of tasks (MFT)

**PROGRAM:**

#include<stdio.h>

#include<math.h>

main()

{

int np,nb,mm,bs,i,j,ps[100],nba[100],ifm[100],sb=0,flag=0;

float x;

clrscr();

printf(“Enter the Memory size”);

scanf(“%d”,&mm);

printf(“Enter the no of Blocks”);

scanf(“%d”,&nb);

printf(“Enter the no of processes”);

scanf(“%d”,&np);

bs=mm/nb;

for(i=1;(i<=np)&&(sb<nb);i++)

{

printf(“Enter the size of p[%d]:”,i);

scanf(“%d”,&ps[i]);

if(ps[i]<=bs)

nba[i]=1;

else

{

x=ps[i]/(float)bs;

nba[i]=(ceil)(x);

}

ifm[i]=nba[i]\*bs-ps[i];

sb=sb+nba[i];

if(sb>nb)

{

i=i-1;

flag=1;

}

}

j=i;

printf(“Process \t Size \t nba \t ifm \n”);

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

printf(“%d \t %d \t %d \t %d \n”, i, ps[i], nba[i], ifm[i]);

if(flag==1)

printf(“Memory space is unavailable”);

getch();

}

**Experiment 5)**

**MVT Program**

4) Write a Program to implement Multiprogramming with a variable number of tasks (MVT)

**Aim:** To implement a program for Multiprogramming with a variable number of tasks (MVT)

**PROGRAM:**

#include<stdio.h>

main()

{

int mm,np,ps[100],rm[100],am=0,flag=0,i,j;

clrscr();

printf(“Enter the memory size”);

scanf(“%d”,&mm);

printf(“enter no of processes”);

scanf(“%d”,&np);

for(i=0;(i<np)&&(am<mm);i++)

{

printf(“Enter the size of p[%d]:”,i+1);

scanf(“%d”,&ps[i]);

am=am+ps[i];

if(am>=mm)

{

flag=1;

break;

}

rm[i]=mm-am;

}

j=i;

printf(“Process \t size \t rm \n”);

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

printf(“%d \t %d \t %d \n ”, i+1, ps[i] , rm[i]);

if(flag==1)

printf(“memory is unavailable”);

getch();

}