CERTIFICATE

Name of the Lab : OPERATING SYSTEMS

Name of the Student : shaik sharmila

Regd. No. : 19BQ5A0528

CLASS : III B.TECH. I SEM CSE – D

GIT HUB LINK:<https://github.com/sharmila528/oslab>

19BQ5A0528 Page 1

INDEX

S. No. Name of the Experiment LAST DATE PAGE NO 3 (a) Simulate Multiprogramming with a fixed number of tasks (MFT) OCT 17 3-9 3 (b) Simulate Multiprogramming with a variable number of tasks

10-13

(MVT)

19BQ5A0528 Page 2

**EXPERIMENT NO: 3 (a)**

**AIM :** Simulate Multiprogramming with a fixed number of tasks (MFT) **DESCRIPTION :**

**LIBRARIES USED:**

**SYNTAX:**

**PROGRAM-1:**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int fragment[20],b[20],p[20],i,j,nb,np,temp,lowest=9999,flags[10],**

**allocation[10],pm,highest=0,temp1;**

**static int barray[20],parray[20],bar[20],par[20];**

**cout<<"Enter the size of physical memory for user processes:";**

**cin>>pm;**

**cout<<"\nBEST FIT";**

**cout<<"\nEnter the number of partitions:";**

**cin>>nb;**

**cout<<"Enter the number of processes:";**

**cin>>np;**

**cout<<"\nEnter the size of the partitions:-\n";**

**for(i=1;i<=nb;i++)**

19BQ5A0528 Page 3

**{**

**cout<<"Partition no."<<i<<":";**

**cin>>b[i];**

**}**

**cout<<"\nEnter the size of the processes :-\n";**

**for(i=1;i<=np;i++)**

**{**

**cout<<"Process no. "<<i<<":";**

**cin>>p[i];**

**}**

**for(i=1;i<=np;i++)**

**{**

**for(j=1;j<=nb;j++)**

**{**

**if(barray[j]!=1)**

**{**

**temp=b[j]-p[i];**

**if(temp>=0)**

**if(lowest>temp)**

**{**

**parray[i]=j;**

**lowest=temp;**

**}**

**}**

19BQ5A0528 Page 4

**}**

**fragment[i]=lowest;**

**barray[parray[i]]=1;**

**lowest=10000;**

**}**

**cout<<"\nProcess\_no\tProcess\_size\tBlock\_no\tBlock\_size\tFragment";**

**for(i=1;i<=np && parray[i]!=0;i++)**

**cout<<"\n"<<i<<"\t\t"<<p[i]<<"\t\t"<<parray[i]<<"\t\t"<<b[parray[i]]<<"\t\t"<<fragment[i]; for(i = 0; i < 10; i++)**

**{**

**flags[i] = 0;**

**allocation[i] = -1;**

**}**

**for(i = 1; i <= np; i++) //allocation as per first fit**

**for(j = 1; j <= nb; j++)**

**if(flags[j] == 0 && b[j] >= p[i])**

**{**

**allocation[j] = i;**

**flags[j] = 1;**

**break;**

**}**

**//display allocation details**

**cout<<"\n\nFIRST FIT";**

**cout<<"\nBlock no.\tsize\t\tprocess no.\t\tsize\t\tInternal-Fragmentation"; for(i = 1; i <=nb; i++)**

19BQ5A0528 Page 5

**{**

**cout<<"\n"<< i<<"\t\t"<<b[i]<<"\t\t";**

**if(flags[i] == 1)**

**cout<<allocation[i]+1<<"\t\t\t"<<p[allocation[i]]<<"\t\t\t"<<b[i]-p[allocation[i]]; else**

**cout<<"Not allocated";**

**}**

**cout<<"\n\nWORST FIT";**

**for(i=1;i<=np;i++)**

**{**

**for(j=1;j<=nb;j++)**

**{**

**if(bar[j]!=1)**

**{**

**temp1=b[j]-p[i];**

**if(temp1>=0)**

**if(highest<temp1)**

**{**

**par[i]=j;**

**highest=temp1;**

**}**

**}**

**}**

**fragment[i]=highest;**

**bar[par[i]]=1;**

19BQ5A0528 Page 6

**highest=0;**

**}**

**cout<<"\nProcess\_no\tProcess\_size\tBlock\_no\tBlock\_size\tFragment";**

**for(i=1;i<=np && par[i]!=0;i++)**

**cout<<"\n"<<i<<"\t\t"<<p[i]<<"\t\t"<<par[i]<<"\t\t"<<b[par[i]]<<"\t\t"<<fragment[i]; return 0;**

**}**

**OUTPUT:**

Enter the size of physical memory for user processes:500

BEST FIT

Enter the number of partitions:4

Enter the number of processes:4

Enter the size of the partitions:-

Partition no.1:100

Partition no.2:80

Partition no.3:120

Partition no.4:200

Enter the size of the processes :-

Process no. 1:70

Process no. 2:90

Process no. 3:110

Process no. 4:150

Process\_no Process\_size Block\_no Block\_size Fragment

1 70 2 80 10

2 90 1 100 10

19BQ5A0528 Page 7

3 110 3 120 10

4 150 4 200 50

FIRST FIT

Block no. size process no. size Internal-Fragmentation 1 100 2 70 30

2 80 Not allocated

3 120 3 90 30

4 200 4 110 90

WORST FIT

Process\_no Process\_size Block\_no Block\_size Fragment

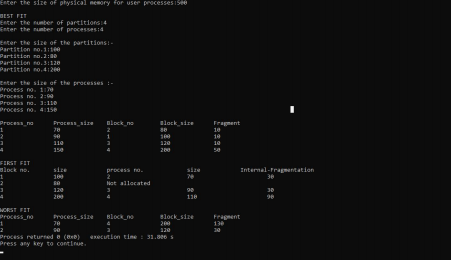
1 70 4 200 130

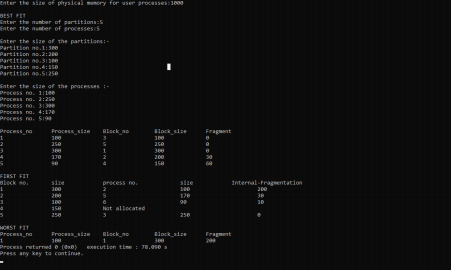
2 90 3 120 30

**OUTPUT SCREEN SHOTS:**

**OUTPUT-1:**

19BQ5A0528 Page 8

**OUTPUT-2:**

****19BQ5A0528 Page 9

**EXPERIMENT NO: 3 (B)**

**AIM :** Simulate Multiprogramming with a variable number of tasks (MVT) **DESCRIPTION :**

**LIBRARIES USED:**

**SYNTAX:**

**PROGRAM-1:**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int ms,mp[10],i, temp,n=0;**

**char ch = 'y';**

**cout<<"\nEnter the size of physical memory for user processes:";**

**cin>>ms;**

**temp=ms;**

**for(i=0;ch=='y';i++,n++)**

**{**

**cout<<"\nEnter memory required for process"<<i+1<<":";**

**cin>>mp[i];**

**if(mp[i]<=temp)**

**{**

19BQ5A0528 Page 10

**cout<<"\nMemory is allocated for Process"<<i+1<<":";**

**temp = temp - mp[i];**

**}**

**else**

**{**

**cout<<"\nMemory is Full";**

**break;**

**}**

**cout<<"\nDo you want to continue(y/n) -- ";**

**cin>>ch;**

**}**

**cout<<"\n\nTotal Memory Available:"<<ms;**

**cout<<"\n\n\tPROCESS\t\t MEMORY ALLOCATED ";**

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

**cout<<"\n \t"<<i+1<<"\t\t"<<mp[i];**

**cout<<"\n\nTotal Memory Allocated is:"<<ms-temp;**

**cout<<"\nTotal External Fragmentation is:"<<temp;**

**}**

**OUTPUT :**

Enter the size of physical memory for user processes:500

Enter memory required for process1:70

Memory is allocated for Process1:

Do you want to continue(y/n) -- y

Enter memory required for process2:90

19BQ5A0528 Page 11

Memory is allocated for Process2:

Do you want to continue(y/n) -- y

Enter memory required for process3:110

Memory is allocated for Process3:

Do you want to continue(y/n) -- y

Enter memory required for process4:150

Memory is allocated for Process4:

Do you want to continue(y/n) -- n

Total Memory Available:500

PROCESS MEMORY ALLOCATED

1 70

2 90

3 110

4 150

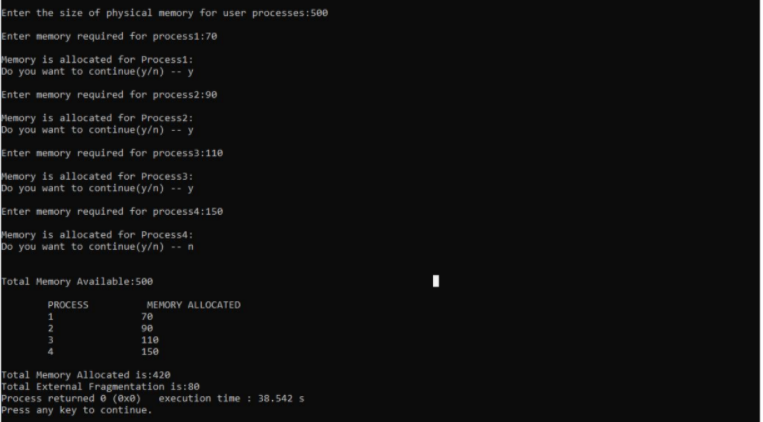
Total Memory Allocated is:420

Total External Fragmentation:80

**OUTPUT SCREEN SHOTS:**

**OUTPUT-1:**

19BQ5A0528 Page 12

OUTPUT-2:

19BQ5A0528 Page 13