# Lab 10

1. WAP to simulate the following contiguous memory allocation technique:
   1. Worst-fit b. Best-fit c. First-fit

# Sample Code:

**Memory Allocation (Worst fit)** #include<stdio.h> #include<conio.h>

#define max 25 int main()

{

int frag[max],b[max],f[max],i,j,nb,nf,temp; static int bf[max],ff[max]; printf("\n\tMemory Management Scheme - Worst 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;

}

}

}

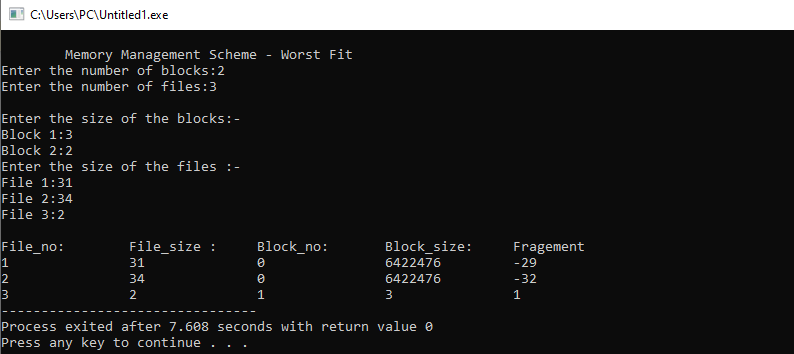
frag[i]=temp; bf[ff[i]]=1;

}

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]);

}

**Output:**

****

**Memory Allocation (Best fit)** #include<stdio.h> #include<conio.h>

#define max 25 int main()

{

int frag[max],b[max],f[max],i,j,nb,nf,temp,lowest=10000; static int bf[max],ff[max];

printf("\n\tMemory Management Scheme - Best 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)

}

if(lowest>temp)

{

ff[i]=j; lowest=temp;

}

}}

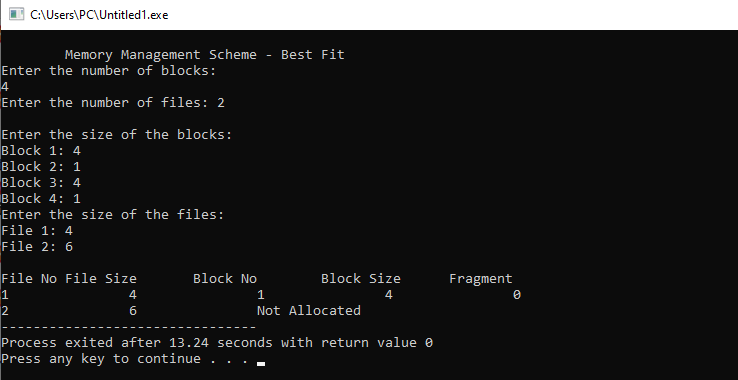
frag[i]=lowest; bf[ff[i]]=1; lowest=10000;

}

printf("\nFile No\tFile Size \tBlock No\tBlockSize\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]);

}

**Output:**

****

**Memory Allocation (First fit)** #include<stdio.h> #include<conio.h>

int main()

{

int bsize[10], psize[10], bno, pno, flags[10], allocation[10], i, j; for(i = 0; i < 10; i++)

{

flags[i] = 0;

allocation[i] = -1;

}

printf("Enter no. of blocks: "); scanf("%d", &bno);

printf("\nEnter size of each block: "); for(i = 0; i < bno; i++)

scanf("%d", &bsize[i]); printf("\nEnter no. of processes: "); scanf("%d", &pno);

printf("\nEnter size of each process: "); for(i = 0; i < pno; i++)

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

for(i = 0; i < pno; i++) //allocation as per first fit for(j = 0; j < bno; j++)

if(flags[j] == 0 && bsize[j] >= psize[i])

{

allocation[j] = i; flags[j] = 1; break;

}

//display allocation details

printf("\nBlock no.\tsize\t\tprocess no.\t\tsize"); for(i = 0; i < bno; i++)

{

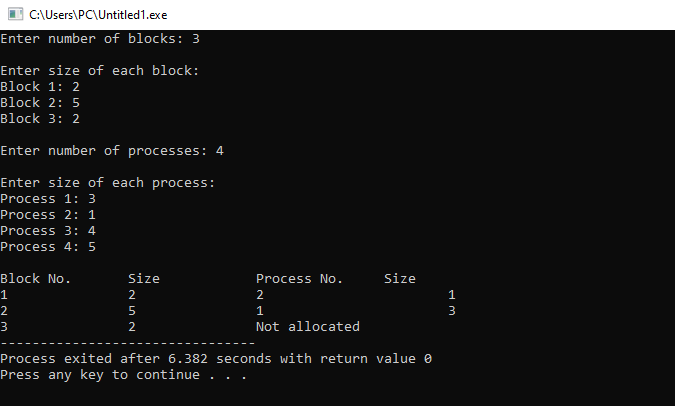
printf("\n%d\t\t%d\t\t", i+1, bsize[i]); if(flags[i] == 1)

printf("%d\t\t\t%d",allocation[i]+1,psize[allocation[i]]);

else

}

}

**Output:**

printf("Not allocated");

1. WAP to implement sequential file allocation technique.

# Sample Code:

// C program to implement sequential file allocation method #include<stdio.h>

#include<conio.h> int main()

{

int f[50], i, st, len, j, c, k, count = 0; for(i=0;i<50;i++)

f[i]=0;

printf("Files Allocated are : \n"); begin:

printf("\n Enter the starting block and lenght of file:"); scanf("%d%d", &st,&len);

for(k=st;k<(st+len);k++) if(f[k]==0)

count++; if(len==count)

{

for(j=st;j<(st+len);j++) if(f[j]==0)

{

f[j]=1;

printf("%d\t%d\n",j,f[j]);

}

if(j!=(st+len-1))

printf("The file is allocated to disk\n");

}

else

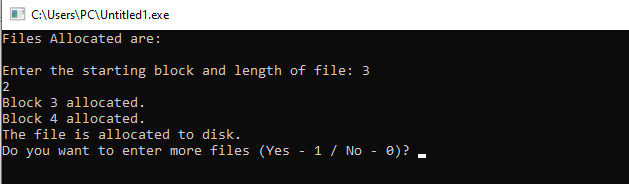
printf(" The file is not allocated \n");

printf("Do you want to enter more file(Yes - 1/No - 0)"); scanf("%d", &c);

if(c==1) goto begin; return 0;

}

**Output:**



1. WAP to implement linked list file allocation technique.

# Sample code:

// C code to implement linked file allocation technique #include<stdio.h>

#include<conio.h> #include<stdlib.h> int main()

{

int f[50], p,i, st, len, j, c, k, a; for(i=0;i<50;i++)

f[i]=0;

printf("Enter how many blocks already allocated: "); scanf("%d",&p);

printf("Enter blocks already allocated: "); for(i=0;i<p;i++)

{

scanf("%d",&a); f[a]=1;

}

x: printf("Enter index starting block and length: "); scanf("%d%d", &st,&len);

k=len; if(f[st]==0)

{

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

{

if(f[j]==0)

{ f[j]=1;

printf("%d >%d\n",j,f[j]);

}

else

{

printf("%d Block is already allocated \n",j); k++;

}

}

}

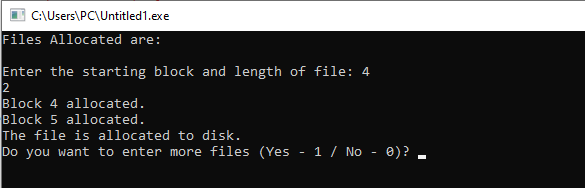
else

printf("%d starting block is already allocated \n",st); printf("Do you want to enter more file(Yes - 1/No - 0)"); scanf("%d", &c);

if(c==1) goto x; else exit(0);

}

**Output:**



1. WAP to implement single level directory.

# Sample code:

//C program to simulate single level directory #include<stdio.h>

#include<conio.h> #include<string.h> int main()

{

int nf=0,i=0,j=0,ch;

char mdname[10],fname[10][10],name[10]; printf("Enter the directory name:"); scanf("%s",mdname);

printf("Enter the number of files:"); scanf("%d",&nf);

do

{

printf("Enter file name to be created:"); scanf("%s",name);

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

{

if(!strcmp(name,fname[i])) break;

}

if(i==nf)

{

strcpy(fname[j++],name); nf++;

}

else

printf("There is already %s\n",name);

printf("Do you want to enter another file(yes - 1 or no - 0):");

scanf("%d",&ch);

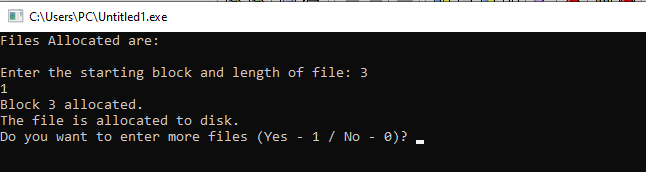
}

while(ch==1);

printf("Directory name is:%s\n",mdname); printf("Files names are:");

for(i=0;i<j;i++) printf("\n%s",fname[i]);

}

**Output:**