# 10. Implementation of File Organization Techniques

**727821TUCS046 DIVYAKARTHIKAA J P**

# CONTIGUOUS (SEQUENTIAL) FILE ALLOCATION ALGORITHM

#include<stdio.h>

#include<conio.h>

main()

{

int n,i,j,b[20],sb[20],t[20],x,c[20][20];

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:”);

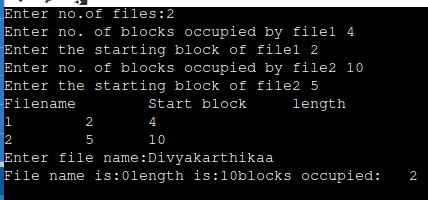
for(i=0;i<b[x-1];i++)

printf(“%4d”,c[x-1][i]);

getch();

}

# Output:



**LINKED FILE ALLOCATION ALGORITHM**

#include<stdio.h>

struct file

{

char fname[10];

int start, size, block[10];

};

int main()

{

struct file f[10];

int i, j, n;

printf("Enter the number 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 number 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("\nFile\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]);

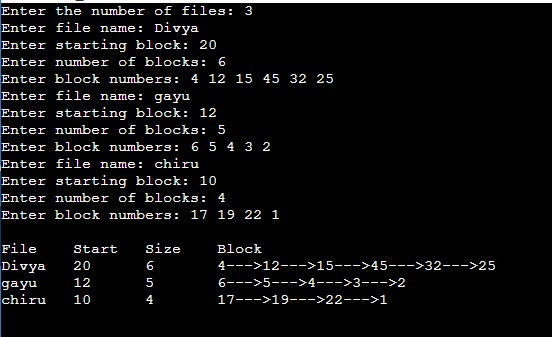
printf("\n");

}

return 0;

}

**OUTPUT:**

****

**INDEXED FILE ALLOCATION ALGORITHM**

#include <stdio.h>

int main()

{

int n, m[20], i, j, sb[20], s[20], b[20][20], x;

printf("Enter the number of files: ");

scanf("%d", &n);

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

{

printf("Enter the starting block and size of file %d: ", i + 1);

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

printf("Enter the number of blocks occupied by file %d: ", i + 1);

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

printf("Enter the blocks of file %d: ", i + 1);

for (j = 0; j < m[i]; j++)

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

}

printf("\nFile\tIndex\tLength\n");

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

{

printf("%d\t%d\t%d\n", i + 1, sb[i], m[i]);

}

printf("\nEnter the file index: ");

scanf("%d", &x);

printf("File index: %d\n", x);

i = x - 1;

printf("Index: %d\n", sb[i]);

printf("Blocks occupied are: ");

for (j = 0; j < m[i]; j++)

printf("%3d", b[i][j]);

return 0;

}

**OUTPUT :**

