**LAB PROGRAMS-4**

**Simulate the following file allocation strategies.**

1. **Sequential**

**Source code:**

#include<stdio.h>

#include<math.h>

struct block{

    int fno;

    int l;

    int st;

}b[10];

void main(){

    int free,n,s,i=0,ch,x,y;

    printf("Enter no of blocks:");

    scanf("%d",&n);

    printf("\nEnter block size:");

    scanf("%d",&s);

    free=n;

    while(1){

        printf("\nEnter File no. & Size\n");

        scanf("%d %d",&x,&y);

        if(free>ceil((double)y/s)){

                b[i].fno=x;

                b[i].l=ceil((double)y/s);

                b[i].st=n-free;

                free=free-b[i].l;

                i++;

        }

        else{

            printf("/Ran out of memory");

            break;

        }

        printf("\nEnter more?(0/1)\n");

        scanf("%d",&ch);

        if(!ch)

            break;

    }

    printf("\nFile Allocation Table\n");

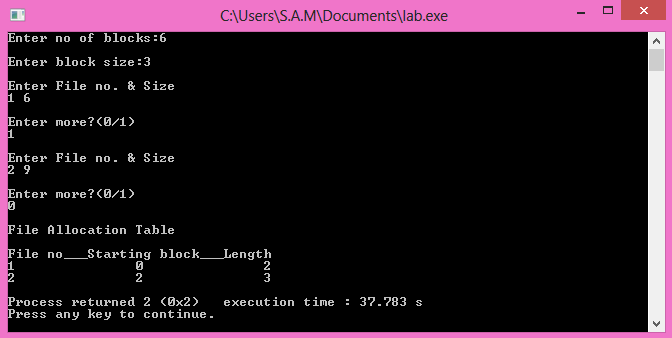
    printf("\nFile no\_\_\_Starting block\_\_\_Length\n");

    for(int j=0;j<i;j++)

        printf("%d\t\t%d\t\t%d\n",b[j].fno,b[j].st,b[j].l);

}

**Output:**



1. **Indexed**

**Source code:**

#include<stdio.h>

struct block{

    int fname;

    int index;

    int l;

    int st;

    int bl;

    int bno[10];

}b[10];

void main(){

    int i,j,x,ch,n;

    printf("Enter no of files:");

    scanf("%d",&n);

    for(i=0;i<n;i++){

        b[i].fname=i+1;

        b[i].index=i;

        printf("\nEnter Starting block & Size of File %d\n",b[i].fname);

        scanf("%d %d",&b[i].st,&b[i].l);

        printf("\nEnter no. of Blocks occupied by File\n");

        scanf("%d",&b[i].bl);

        printf("\nEnter Blocks of File\n");

        for(j=0;j<b[i].bl;j++)

            scanf("%d",&b[i].bno[j]);

    }

    printf("\nFile Allocation Table\n");

    printf("\nFile name\_\_\_\_\_\_Index\_\_\_\_\_\_\_\_Length\n");

    for(j=0;j<n;j++)

        printf("%d\t\t%d\t\t%d\n",b[j].fname,b[j].index,b[j].bl);

    while(1){

    printf("Check for files?(0/1)");

    scanf("%d",&ch);

    if(ch){

        printf("\nEnter File name:");

        scanf("%d",&x);

        i=x-1;

        if(i<0||i>=n){

            printf("\nNo such file exists");

            break;

        }

        printf("Index is:%d\n",b[i].index);

        printf("Blocks occupied are:");

        for(j=0;j<b[i].bl;j++)

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

        printf("\n");

        }

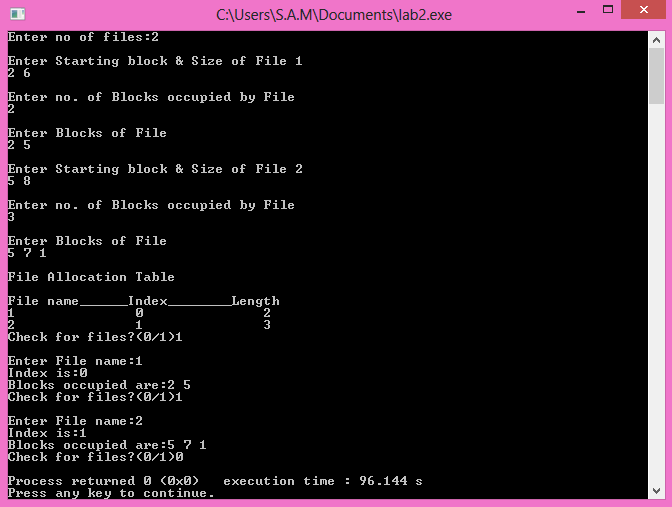
    else

        break;

    }

}

**Output:**

****

**3. Linked**

**Source code:**

#include<stdio.h>

struct block{

    int fname;

    int st;

    int bl;

    int bno[10];

}b[10];

void main(){

    int i,j,x,ch,n;

    printf("Enter no of files:");

    scanf("%d",&n);

    for(i=0;i<n;i++){

        b[i].fname=i+1;

        printf("\nEnter Starting block of File %d:",b[i].fname);

        scanf("%d",&b[i].st);

        printf("\nEnter no. of Blocks occupied by File:");

        scanf("%d",&b[i].bl);

        b[i].bno[0]=b[i].st;

        printf("\nEnter Blocks of File\n");

        for(j=1;j<=b[i].bl-1;j++)

            scanf("%d",&b[i].bno[j]);

    }

    printf("\nFile Allocation Table\n");

    printf("\nFile name\_\_\_\_\_\_Start\_\_\_\_\_\_\_\_\_\_Size\_\_\_\_Block\n");

    for(i=0;i<n;i++){

        printf("%d\t\t%d\t\t%d\t",b[i].fname,b[i].st,b[i].bl);

        for(j=0;j<b[i].bl-1;j++)

            printf("%d-->",b[i].bno[j]);

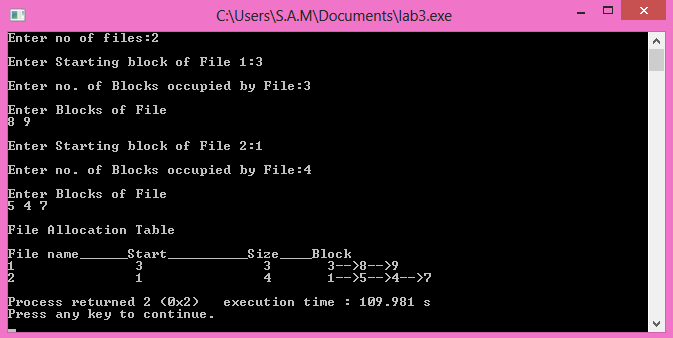
        printf("%d",b[i].bno[j]);

        printf("\n");

    }

}

**Output:**

****

**SUBMITTED BY**

**ARJUN TS**

**CHN18CS026**