

Experiment No. 4

4. Implement all file allocation strategies:

Sequential

Indexed

Linked

DESCRIPTION

A file is a collection of data, usually stored on disk. As a logical entity, a file enables to divide data into meaningful groups. As a physical entity, a file should be considered in terms of its organization. The term "file organization" refers to the way in which data is stored in a file and, consequently, the method(s) by which it can be accessed.

SEQUENTIAL FILE ALLOCATION

In this file organization, the records of the file are stored one after another both physically and logically. That is, record with sequence number 16 is located just after the 15th record. A record of a sequential file can only be accessed by reading all the previous records.

LINKED FILE ALLOCATION

With linked allocation, each file is a linked list of disk blocks; the disk blocks may be scattered anywhere on the disk. The directory contains a pointer to the first and last blocks of the file. Each block contains a pointer to the next block.

INDEXED FILE ALLOCATION

Indexed file allocation strategy brings all the pointers together into one location: an index block. Each file has its own index block, which is an array of disk-block addresses. The i^{th} entry in the index block points to the i^{th} block of the file. The directory contains the address of the index block. To find and read the i^{th} block, the pointer in the i^{th} index-block entry is used.

SEQUENTIAL FILE ALLOCATION:

```
#include<stdio.h>
```

```
struct fileTable {  
    char name[20];  
    int sb, nob;  
    }ft[30];
```

```
void main() {  
    int i, j, n; char s[20];  
    printf("Enter no of files :");  
    scanf("%d",&n);  
  
    for(i=0;i<n;i++) {
```

```

printf("\nEnter file name %d :",i+1);
scanf("%s",ft[i].name);
printf("Enter starting block of file %d :",i+1);
scanf("%d",&ft[i].sb);
printf("Enter no of blocks in file %d :",i+1);
scanf("%d",&ft[i].nob);
}

printf("\nEnter the file name to be searched -- ");
scanf("%s",s);
for(i=0;i<n;i++)
    if(strcmp(s, ft[i].name)==0)
        break;
if(i==n)
    printf("\nFile Not Found");
else {
    printf("\nFILE NAME \tSTART BLOCK \tNO OF BLOCKS \tBLOCKS
OCCUPIED\n");
    printf("\n%s\t\t%d\t\t%d\t\t",ft[i].name,ft[i].sb,ft[i].nob);
    for(j=0;j<ft[i].nob;j++)
        printf("%d, ",ft[i].sb+j);
    }
}
}

```

Output:

```

C:\Users\nouma\Documents\Programs\ifa.exe
Enter no of files :3
Enter file name 1 :A
Enter starting block of file 1 :85
Enter no of blocks in file 1 :6
Enter file name 2 :B
Enter starting block of file 2 :60
Enter no of blocks in file 2 :4
Enter file name 3 :C
Enter starting block of file 3 :120
Enter no of blocks in file 3 :4
Enter the file name to be searched -- B
FILE NAME      START BLOCK    NO OF BLOCKS   BLOCKS OCCUPIED
B              60             4              60, 61, 62, 63,
Process returned 4 (0x4)   execution time : 20.234 s
Press any key to continue.

```

LINKED FILE ALLOCATION:

```
#include<stdio.h>
```

```
struct fileTable
{
    char name[20];
    int nob;
    struct block *sb;
}ft[30];
```

```
struct block
{
    int bno;
    struct block *next;
};
```

```
void main()
{
    int i, j, n;
    char s[20];
    struct block *temp;
    printf("Enter no of files :");
    scanf("%d",&n);

    for(i=0;i<n;i++)
    {
        printf("\nEnter file name %d :",i+1);
        scanf("%s",ft[i].name);
        printf("Enter no of blocks in file %d :",i+1);
        scanf("%d",&ft[i].nob);
        ft[i].sb=(struct block*)malloc(sizeof(struct block));
        temp = ft[i].sb;
        printf("Enter the blocks of the file :");
        scanf("%d",&temp->bno);
        temp->next=NULL;

        for(j=1;j<ft[i].nob;j++)
        {
            temp->next = (struct block*)malloc(sizeof(struct block));
            temp = temp->next;
            scanf("%d",&temp->bno);
        }

        temp->next = NULL;
    }
}
```

```

printf("\nEnter the file name to be searched -- ");
scanf("%s",s);

for(i=0;i<n;i++)
    if(strcmp(s, ft[i].name)==0)
        break;
    if(i==n)
        printf("\nFile Not Found");
    else
    {
        printf("\nFILE NAME \tNO OF BLOCKS \tBLOCKS OCCUPIED");
        printf("\n %s\t\t%d\t\t",ft[i].name,ft[i].nob);
        temp=ft[i].sb;

        for(j=0;j<ft[i].nob;j++)
        {
            printf("%d --> ",temp->bno);
            temp = temp->next;
        }
    }
}

```

Output:

```

C:\Users\nouma\Documents\Programs\lfa.exe
Enter no of files :2
Enter file name 1 :A
Enter no of blocks in file 1 :4
Enter the blocks of the file :11 22 33 44

Enter file name 2 :B
Enter no of blocks in file 2 :2
Enter the blocks of the file :10 20

Enter the file name to be searched -- B

FILE NAME      NO OF BLOCKS    BLOCKS OCCUPIED
B              2              10 --> 20 -->
Process returned 2 (0x2)   execution time : 37.527 s
Press any key to continue.

```

INDEXED FILE ALLOCATION:

```
#include<stdio.h>

struct fileTable
{
    char name[20];
    int nob,
    blocks[30];
}ft[30];

void main()
{
    int i, j, n;
    char s[20];
    printf("Enter no of files :");
    scanf("%d",&n);

    for(i=0;i<n;i++)
    {
        printf("\nEnter file name %d :",i+1);
        scanf("%s",ft[i].name);
        printf("Enter no of blocks in file %d :",i+1);
        scanf("%d",&ft[i].nob);
        printf("Enter the blocks of the file :");

        for(j=0;j<ft[i].nob;j++)
            scanf("%d",&ft[i].blocks[j]);
    }

    printf("\nEnter the file name to be searched -- ");
    scanf("%s",s);

    for(i=0;i<n;i++)
        if(strcmp(s, ft[i].name)==0)
            break;
    if(i==n)
        printf("\nFile Not Found");
    else
    {
        printf("\nFILE NAME \tNO OF BLOCKS \tBLOCKS OCCUPIED");
        printf("\n %s\t\t%d\t\t",ft[i].name,ft[i].nob);

        for(j=0;j<ft[i].nob;j++)
            printf("%d, ",ft[i].blocks[j]);
    }
}
```

Output:

```
C:\Users\nouma\Documents\Programs\indexed.exe
Enter no of files :2
Enter file name 1 :A
Enter no of blocks in file 1 :3
Enter the blocks of the file :11 22 33
Enter file name 2 :B
Enter no of blocks in file 2 :6
Enter the blocks of the file :10 20 30 40 50 60
Enter the file name to be searched -- B
FILE NAME      NO OF BLOCKS    BLOCKS OCCUPIED
B              6              10, 20, 30, 40, 50, 60,
Process returned 6 (0x6)   execution time : 141.661 s
Press any key to continue.
```