

Experiment-35: Consider a file system that brings all the file pointers together into an index block. The I^{th} entry in the index block points to the i^{th} block of the file. Design a C program to simulate the file allocation strategy.

Aim:

To simulate the file allocation strategy where all file pointers are brought together into an index block, and the i^{th} entry in the index block points to the i^{th} block of the file.

Procedure:

1. Take the number of file blocks as input.
2. Use an index block to store pointers to the actual blocks in the file.
3. Access each block through the index block and display the content of the block.

C Program:

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

```
int main() {
```

```
    int n;
```

```
    printf("Enter the number of file blocks: ");
```

```
    scanf("%d", &n);
```

```
    int index_block[n], file_blocks[n];
```

```
    printf("Enter the data for the file blocks:\n");
```

```
    for (int i = 0; i < n; i++) {
```

```
        printf("Block %d data: ", i + 1);
```

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

```
    }
```

```
    for (int i = 0; i < n; i++) {
```

```
        index_block[i] = i;
```

```
    }
```

```
printf("Enter the block number to access (1 to %d): ", n);

int block;

scanf("%d", &block);

if (block < 1 || block > n) {

    printf("Invalid block number.\n");

} else {

    printf("Accessing Block %d: %d\n", block, file_blocks[index_block[block - 1]]);

}

return 0;

}
```

Output:

Output

```
Enter the number of file blocks: 4
Enter the data for the file blocks:
Block 1 data: 5
Block 2 data: 2
Block 3 data: 3
Block 4 data: 1
Enter the block number to access (1 to 4): 2
Accessing Block 2: 2
```

```
=== Code Execution Successful ===192372048
```