

EXPERIMENT 14

FILE ALLOCATION STRATEGY

AIM

Write a C program to simulate following file allocation strategies
Sequential, Indexed & Linked

SEQUENTIAL ALGORITHM

STEP 0: START

STEP 1: Define a structure named 'file' with members 'name', 'num' and 'start'. Declare 'a' as array of structures 'file', 'n', 'i', 'j' and 'name'.

STEP 2: Prompt the user to enter the number of files (n)

STEP 3: For each file $i = 0$ till $n-1$, prompt user to enter the name, starting block and number of blocks for the file. Read & store the details in array 'a'.

STEP 4: Prompt user to enter the name of file to be searched ('nam')

STEP 5: Iterate through the array 'a' to find file with matching name 'nam'.

STEP 6: Print the details of the found file - Name, Start Block, No of Blocks.

STEP 7: Display the number of blocks occupied by incrementing start block.

INDEXED ALGORITHM

STEP 0: START

STEP 1: Define a structure named 'file' with members 'name', 'num' and an array 'block' to store block numbers

STEP 2: Declare variables 'a' as an array of structures 'file', 'n', 'i', 'j' and 'nam'.

STEP 3: Prompt the user to enter the number of files (n)

STEP 4: From $i=0$ to $n-1$, Prompt the user to enter the name and the number of blocks for the file and store it in array 'a'.

STEP 5: For each file i , Prompt user to enter the block occupied by the file. Read & store each block number in the 'block' array.

STEP 6: Prompt the user to enter the name of file to be searched ('nam')

STEP 7: Iterate through array 'a' to find the file to be searched 'nam'.

STEP 8: Print the details of the found file
- Name, No of Blocks and Blocks Occupied.

LINKED FILE ALLOCATION ALGORITHM

STEP 0: START

STEP 1: Define two structures 'block' for individual blocks and 'file' for file information.

STEP 2: Declare variables including arrays 'a' & 'head', integers n, i, j and character array 'nam'.

STEP 3: Initialize head pointers 'head' to NULL for each file.

STEP 4: Prompt user to enter the no. of files (n).

STEP 5: From file $i = 0$ till $n - 1$. Initialize head pointer to NULL. Prompt user to enter name & number of blocks in the file. Read & store the details in array 'a'.

STEP 6: For each file i - Prompt user to input blocks occupied by the file. Allocate memory for each block, store the block number, and link the block in linked list structure.

STEP 7: Prompt user to enter the name of file to be searched (nam).

Iterate through array 'a' to find the file matching name 'nam'.

STEP 8: Print the details of the file that is found - Name, Number of blocks and Blocks Occupied by

traversing the list.

RESULT

Experiment executed successfully
& output obtained.