

GROUP F Assignment 11

Problem statement:

Department maintains a student information. The file contains roll number, name, division and address. Allow user to add, delete information of student. Display information of particular employee. If record of student does not exist an appropriate message is displayed. If it is, then the system displays the student details. Use sequential file to main the data

Prerequisite:

1. Knowledge of the Object Oriented Language like C++.
2. Concept of dynamic allocation.

Theory:

File structure–

A file is a collection of records which are related to each other. The size of file is limited by the size of memory and storage medium.

Two characteristics determine how the file is organised:

I. File Activity:

It specifies that percent of actual records proceeds in single run. If a small percent of record is accessed at any given time, the file should be organized on disk for the direct access in contrast.

If a fare percentage of records affected regularly than storing the file on tape would be more efficient & less costly.

II. File Volatility:

It addresses the properties of record changes. File records with many changes are highly volatile means the disk design will be more efficient than tape.

File organisation –

A file is organised to ensure that records are available for processing. It should be designed with the activity and volatility information and the nature of storage media, Other consideration are cost of file media, enquiry, requirements of users and file's privacy, integrity, security and confidentiality.

There are four methods for organising files-

1. Sequential organisation
2. Indexed Sequential organisation

3. Inverted list organization
4. Direct access organisation
5. Chaining

1. Sequential organization:

Sequential organization means storing and sorting in physical, contiguous blocks within files on tape or disk. Records are also in sequence within each block. To access a record previous records within

the block are scanned. In a sequential organization, records can be added only at the end of the file. It is not possible to insert a record in the middle of the file without rewriting the file.

In a sequential file update, transaction records are in the same sequence as in the master file. Records from both the files are matched, one record at a time, resulting in an updated master file. In a

personal computer with two disk drives, the master file is loaded on a diskette into drive A, while the transaction file is loaded on another diskette into drive B. Updating the master file transfers data from drive B to A controlled by the software in memory.

Advantages:

- i. Simple to design
- ii. Easy to program
- iii. Variable length and blocked records available
- iv. Best use of storage space

Disadvantages

- i. Records cannot be added at the middle of the file.

2. Indexed sequential organization:

Like sequential organization, keyed sequential organization stores data in physically contiguous blocks. The difference is in the use of indexes to locate records. There are three areas in disk storage: prime area, overflow area and index area. The prime area contains file records stored by key or id numbers. All records are initially stored in the prime area. The overflow area contains records added to the file that cannot be placed in logical sequence in the prime area. The index area is more like a data dictionary. It contains keys of records and their locations on the disk. A pointer associated with each key is an address that tells the system where to find a record.

Advantages:

- i. Indexed sequential organization reduces the magnitude of the sequential search and provides quick access for sequential and direct processing.
- ii Records can be inserted in the middle of the file.

Disadvantages:

- i.It takes longer to search the index for data access or retrieval.
- ii.Unique keys are required.
- iii.Periodic reorganization is require

3. Inverted list organization:

Like the indexed- sequential storage method the inverted list organization maintains an index. The two methods differ, however, in the index level and record storage. The indexed sequential method has a multiple index for a given key, where as the inverted list method has a single index for each key type. In an inverted list, records are not necessarily stored in a particular sequence. They are placed in the

data storage area, but indexes are updated for the record key and location. The inverted keys are best for applications that request specific data on multiple keys. They are ideal for static files because additions and deletions cause expensive pointer updating.

Advantages

- i. Used in applications requesting specific data on multiple keys.

4. Direct access organization:

In direct access file organization, records are placed randomly throughout the file. Records need not be in sequence because they are updated directly and rewritten back in the same location. New records are added at the end of the file or inserted in specific locations based on software commands.

Records are accessed by addresses that specify their disk locations. An address is required for locating a record, for linking records, or for establishing relationships. Addresses are of two types:

- i. Absolute
- ii. Relative

A absolute address represents the physical location of the record. It is usually stated in the format of sector/track/record number. One problem with absolute address is that they become invalid when the file that contains the records is relocated on the disk.

A relative address gives a record location relative to the beginning of the file. There must be fixed length records for reference. Another way of locating a record is by the number of bytes it is from the beginning of the file. When the file is moved, pointers need not be updated because the relative location remains the same.

Advantages:

- i. Records can be inserted or updated in the middle of the file.
- ii. Better control over record allocation.

Disadvantages:

- i. Calculating address required for processing.
- ii. Impossible to process variable length records.

5. Chaining:

File organization requires that relationships be established among data items. It must show how characters form fields, fields form files and files relate to each other. Establishing relationship is done through chaining. It uses pointers.

Example: The file below contains auto parts that are an indexed sequential file sequenced by part no. A record can be retrieved by part no. To retrieve the next record, the whole file has to be searched. This can be avoided by the use of pointers.

Algorithm:

Step 1: Start the Program

Step 2: Obtain the required data through char and int datatypes.

Step 3: Enter the filename, index block.

Step 4: Print the file name index loop.

Step 5: Fill is allocated to the unused index blocks

Step 6: This is allocated to the unused linked allocation.

Step 7: Stop the execution.

Conclusion: In this way we have implemented sequential file in cpp.