Date of submission : 27/03/2020

Shreya Shetty
A033
70362019057.

— PSA Assignment 02 —

Q1. Explain Indexed sequential access scheme for file organisation.

A. Indexed sequential file organisation stores data for fast retrieval. The records in an indexed sequential file are of fixed length and every record is uniquely identified by a key field. We maintain a table known as index table which stores the record number and addresses of all the records. That is for every file. This type of file organisation is called indexed sequential file organisation because physically the records may be stored anywhere but the index table stores the addresses of those records.

The i^{th} entry in the index table points to the i^{th} record of the file. Initially when the file is created, each entry in the index table contains NULL.

When the i^{th} record of the file is written, free space is obtained from free space manager and its address is stored in the i^{th} location of the index table.

RECORD NUMBER	ADDRESS OF THE RECORD
1	765
2	27
3	876
4	742
5	NULL
6	NULL
7	NULL
8	NULL
9	NULL

Now, if one has to find the fourth record there is no need to access the first three records. Address of the fourth record can be obtained from the index table and the record can be straight away read from the specified address (742, acc to the given example).

An Indexed sequential file uses the concept of both sequential as well as

relative files. While the index table is read sequentially to find addresses of the desired record a direct access is made to the address of the specified record in order to access it randomly

Indexed sequential files perform well in situations where sequential access as well as random access is made to the data. Indexed sequential files can be stored only on devices that supports random access for example, magnetic discs.

ADVANTAGES OF INDEXED SEQUENTIAL FILE ORGANISATION :

1. The key improvement is that the indices are as small and can be searched quickly, allowing the data base to access only the records it needs.
2. supports applications that require both batch and interactive processing.
3. Records can be accessed sequentially as well as randomly
4. Updates the records in the same file.

DISADVANTAGES OF INDEXED SEQUENTIAL FILE ORGANISATION:

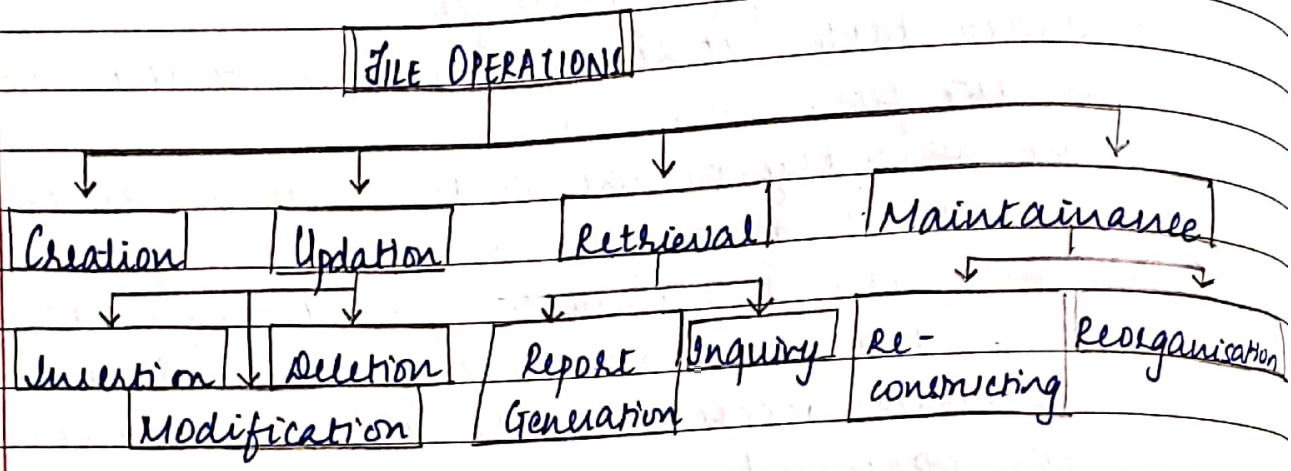
1. Indexed sequential files can be stored only on disks.
2. Needs extra space and overhead to store indices
3. Handling these files is more complicated than handling sequential files.
4. supports only fixed length records.

BASIC FEATURES:

1. Provides fast data retrieval.
2. Records are of fixed length
3. Index table stores the address of the records in the file.
4. The ith entry in the index table points to the ith record of the file.
5. While the Index table is read sequentially to find the address of the desired record, a direct access is made to the address of the specified record in order to access it randomly.
6. Indexed sequential files perform well in situations where sequential access as well as random access is made to the data.

Q2. What are basic file operations?

A.



CREATING A FILE:

A file is created by specifying its name and mode, then the file is opened for writing records that are read from an input device. Once all the records have been written into the file the file is closed. The file is now available for future read/write operations by any program that has been designed to use in some way or another.

UPDATING A FILE:

Updating a file means changing the content of the file to reflect current picture of reality. A file can be updated in the following ways:

- Inserting a new record in the file.
- Deleting an existing record
- Modifying an existing record.

eg.

→ Insertion: If a new student joins the course, we need to add his record to the STUDENT file.

→ Deletion: If a student quits a course in the middle of the session, his record has to be deleted from the STUDENT file.

→ Modification: If the name of a student was spelt incorrectly the correcting the name will be modification of the existing record.

RETRIEVING FROM A FILE

It means extracting useful data from a given file. Information can be retrived from a file either for an inquiry or for report generation. An inquiry for some data retrieves low volume of data, while report generation may retrieve a large volume of data from the file.

MAINTAINING A FILE

It involves reconstruction or reorganising the file to improve the performance of the programs that access this file.
Reconstructing a file keeps the file organisation unchanged and change only the structural aspects of the file (Eg: Changing the feild width or adding/deleting feilds) On the other hand, File reorganisation may involve ~~the~~ changing the

entire organisation of file.

Q3. Explain Hashing Indices.

A. The concept of hashing can be used to create hashing indices. Hashing is used to compute the address of record by using hash function on the search key value. If at any point of time, the hashed values map to the same address then collision occurs and schemes to resolve these collisions are applied to generate a new address. Choosing a good hash function is critical to the success of this technique. Firstly a good hash function, irrespective of the number of search keys, gives an average case look up that is a small constant. Secondly, the function distributes records uniformly and randomly among the buckets, where a bucket is defined as a unit of one or more records (typically a disk block). Correspondingly, the worst hash function is one that maps all the keys to the same bucket.

DRAWBACKS OF USING OF HASHED INDICES:

- Though the number of buckets is fixed, the number of files may grow with time.
- If the number of buckets is too large, storage space is wasted.
- If the number of buckets is too small, there may be too many collisions.

It is recommended to set the number of buckets to twice the number of the search key values in the file.

A hashed file organisation ~~requires~~ uses hashed indices. Hashing is used to calculate the address of the disk block where the desired record is stored.

If K is the set of all search keys values and B is the set of all bucket addresses then hash function (H) maps K to B.

Insertion, search and deletion are a few operations that can be performed in a hashed file organisation further, the secondary indices need to be organised using hashing.

— x —