A-11

Varun K A016 70362019028

Q1. - Indexed sequential access file combines both both sequential file and direct access file organisation. In this the record are stored randomly on a direct access device such as magnetic disk by a primary key. This file have multiple keys. This keys can be alphanumeric in which the record are ordered is called as primary key. In this the data can be accessed either sequentially or randomly using the index. The index is stored in a file and read into memory when the file is opened.

Q2. There are 6 basic file operations:-

1. Creating A File: There are 2 steps necessary for creating a file. First, space in the file system must be found for the file. We discuss how to allocate space for the file. Second, an entry for the new file must be made in the directory.

2. Writing A File: To write a file, you make a system call specifying both the name of the file along with the information to be written to the file.

3. Reading A File: To read from a file, you use a system call which specifies the name of the file and where within the memory the next block of file should be placed.

4. Repositioning Inside A File: The directory is then searched for suitable entry, and then 'current-file position' pointer is relocated to a given value.

5. Deleting A File: For deleting a file, we have to search the directory for specific file.

Varun K A016 70382019028 A-11

Page No.
Date

6. Truncating A File: The user may wish to delete the the entire data but keeping the attributes same.

Q3. So far, we have studied that hashing is used to compute the address of a record by using a 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 this collision are applied to generate a new address.

Choosing a good hash function is critical to the success of this technique. By a good hash function, we mean 2 things. First, a good hash function, irrespective of the number of search keys, gives an average case lookup 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. Correspondingly, the worst hash function is one that maps all keys to the same bucket. However, the drawback of using hash indices includes :-

- Though the numbers 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 keys values in the file. This gives a good space-performance trade-off.

Varun K A016 70362019028 A-11

Q1 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 feild. We maintain a table known as the INDEX TABLE which stores the record number and the address of all the records. That is for every file, we have an index table.
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. An indexed sequential file uses the concept of both sequential as well as relative file. While that index table is read sequentially to find address of the desired record a direct access is made to the address of the specified record in order to access it randomly. Indexed sequential file perform well in situation where sequential access as well as random access is made to the data. Indexed sequential file can be stored only on desires that support random access.

ADVANTAGES

- Indices are small and can be searched quickly, allowing the data base to access only the records in it.
- Supports application that require both batch and interactive processing.
- Records can be accessed sequentially as well as randomly.
- Updates the records in the same file.

Varun K A016 70362019028 A-11

## DISADVANTAGES

- Can be stored only on disks.
- Needs extra space and overhead to store indices
- Handling is more complicated than sequential file.
- Supports only fixed length records.

## BASIC FEATURES

- Provides fast data retrieval
- Records are of fixed length.
- The $i^{th}$ entry in the index table points to the $i^{th}$ record of the file.
- Index table stores the address of the records in the file.

Eg: If one has to find the 3rd record there is no need to access first 2 records. Address of the 3rd record can be obtained from the index table and record can be straight away read from the specified address [876 according to given example.]

| RECORD NO. | ADDRESS OF THE REC |
|---|---|
| 1 | 765 |
| 2 | 27 |
| 3 | 876 |
| 4 | 742 |
| 5 | NULL |
| 6 | NULL |
| 7 | NULL |
| 8 | NULL |
| 9 | NULL |