Assignment No 11

Problem Statement:

Company maintains employee information as employee ID, name, designation and salary. Allow user to add, delete information of employee. Display information of particular employee. If employee does not exist an appropriate message is displayed. If it is, then the system displays the employee details. Use index sequential file to maintain the data

Theory:

What is the Indexed sequential access method (ISAM)?

ISAM (Indexed sequential access method) is an advanced sequential file organization method. In this case, records are stored in the file with the help of the primary key. For each primary key, an index value is created and mapped to the record. This index contains the address of the record in the file.

If a record has to be obtained based on its index value, the data block's address is retrieved, and the record is retrieved from memory.

Pros of ISAM

- Because each record consists of the address of its data block in this manner, finding a record in a large database is rapid and simple.
- Range retrieval and partial record retrieval are both supported by this approach. We may obtain data for a specific range of values because the index is based on primary key values. Similarly, the partial value can be simply found, for example, in a student's name that begins with the letter 'JA'.

Cons of ISAM

- This approach necessitates additional disc space to hold the index value.
- When new records are added, these files must be reconstructed in order to keep the sequence.
- When a record is erased, the space it occupied must be freed up. Otherwise, the database's performance will suffer.

Algorithm for different operations of Index sequential file:

Algorithm for Creating Sequential file and index file

1. Open Sequential file (emp.DAT) and Index file (ind.DAT) in read mode
2. Input details of employee(Empid, Name and Salary) and store it in record variable.
3. Write record in emp.DAT
4. Copy employee_id and position of record from emp.DAT file into index file.
5. Repeat steps 2,3,4 for all records .
6. Close file pointers for emp.DAT and ind.DAT
7. Stop.

Algorithm for Displaying records from sequential File.

1. Open both files(EMP.DAT and IND.DAT) in read mode
2. Set file pointers for both files at the beginning.
3. Read record from index file.
4. Calculate offset of same record in sequential files .
5. Read record from sequential based on calculated offset.
6. Print record on screen.
7. Repeat steps3,4,5,6 for all records .

Algorithm for deleteing record from file(Logical Deletion)

1. Ask for employee id of a record to be deleted.
2. Get employee id in variable id
3. Open both files (IND.DAT and EMP.DAT) for read and write purpose.
4. Set file pointers at the beginning.
5. Read record .
6. If employee if of record is not -1 then check if employee id of the record =id to be deleted.
7. If id matches then set employee name = " " and employee id and Employee salary=-1 to indicate logical deletion of a record.
8. Close file pointers
9. Stop.

Algorithm to search record

1. Ask for employee id of a record to be searched.
2. Get employee id in variable id
3. Open both files (IND.DAT and EMP.DAT) for read purpose.
4. Set file pointers at the beginning.
5. Read record .
6. If employee if of record is not -1 then check if employee id of the record =id to be searched.
7. If id matches then Display details of employee from sequential file .
8. If record not found till end or employee id for a record is -1 then display message as "Record not found"

Conclusion:

Hence we studied how to Implement indexing for sequential file to decrease search time for a record.