Section 13.2 - File Organisation and Access

Layer 6: High-Order Language

Syllabus Content Section 13: Data Representation

S13.2.1 Show understanding of the methods of file organisation and select an appropriate method of file organisation and file access for a given problem

- Including serial, sequential (using a key field), random (using a record key)

> Serial file organisation

Method of file organisation in which records of data are physically stored in a file, one after another, **in the order they were added to the file**

New records are appended to the end of the file. Serial file organisation is often used for temporary files storing transactions to be made to more permanent files. e.g. storing customer meter reading for gas/electricity before they are used to send the bills to all customers.

- Each transaction is added to the file in the order of arrival
- These records will be in chronological order
- High hit rate
- Suitable for batch processing
- Order not important
- The records will be accessed one after another

> Sequential file organisation

Method of file organisation in which records of data are physically stored in a file, one after another, **in a given order**

Physically stores records of data in a file, one after another, in a given order. The order is usually based on the **key field** of the records as this is the unique identifier.

- High hit rate
- Suitable for batch processing of records
- File organised using unique ID called **key field**

The difference between key fields in a file and primary keys in a database table. In the database table the primary key values must all be unique. This is not a requirement for key fields in any type of file. It may be sensible in certain applications to ensure key fields have unique values, but it is not mandatory.

> Random file organisation

Method of file organisation in which records of data are physically stored in a file in **any available** position, the position of any record in a field is found by applying a **hash algorithm** on the **key field** of the record/record key.

- Records can be added to any empty position
- Key field is hashed to produce the home location
- If home location is free, insert the data
- Else, use overflow method to find free location to store the data
- If no free location is available, the file is full and data cannot be added/stored
- Real-time processing
- Fast access to data
- No need to search through records

S13.2.2 Show understanding of methods of file access

Including

- Sequential access for serial and sequential files
- Direct access for sequential and random files

> File access

Method used to physically find a record in a file

> Sequential access

A method of file access in which records are searched one after another from the physical start of the file until the required record is found, or a new record can be added to the file. This method is used for `serial` and `sequential` files.

- High hit rate

For a `serial` file, if a particular record is being searched for, every record needs to be checked until the record is found or the whole file has been searched and that record has not been found. Any new records are appended to the end of the file.

For a `sequential`file, if a particular records is being searched for, every records needs to be checked until the record is found or *the key filed of the current record being checked is greater than the key field of the record being searched for*. The rest of the file does not need to be searched as the records are sorted on ascending key field values.

Sequential access if efficient when **every** record in the file needs to be processed. e.g. a monthly billing or payroll system.

> Direct access

A method of file access in which a record can be physically found in a file without reading other records.

- Both `sequential` and `random` files can use direct access.
- Allow specific records to be found more quickly than using sequential access.
- Low hit rate

Direct access is required when an **individual** record from a file needs to be processed. e.g. when a single customer record needs to be updated when the customer's phone number is changed.

For a `sequential` file, an index of all the key fields is kept and used to look up the address of the file location where a given record is stored. For large files, searching the index takes less time than searching the whole file.

A separate index file is created which has two fields per record. The first record has the key field value and the second filed has a value for the position of this key field value in the main file.

For a `random access` file, a hashing algorithm is used on the key filed to calculate the address of the file location where a given record is stored.

S13.2.3 Show understanding of hashing algorithms

- Describe and use different hashing algorithms to read from and write data to a random/sequential file

---

> Hashing algorithms

A mathematical formula used to perform a calculation on the `record key`. The result of the calculation gives the address where the record should be found. More complex hashing algorithms are used in the encryption of data.

> Two ways of dealing with collision:

- `open hash`: the record is stored in the next free space.
- `close hash`: an overflow area is set up and the record is stored in the next free space in the overflow area.