Processing Sequential Files

Introduction

Aims

The records in a Sequential file are organized serially, one after another, but the records in the file may be ordered or unordered. The serial organization of the file and whether the file is ordered or unordered has a significant baring on how we process the records in the file and what kind of processing we can do.

This tutorial examines the consequences of ordering and organization and reviews some techniques for processing Sequential files.

Objectives

By the end of this unit you should -

1. Understand how Sequential files are organized.
2. Understand the processing limitations imposed by an unordered Sequential file
3. Be able to add, delete and update records in an ordered Sequential file.

Prerequisites

Introduction to COBOL

Declaring data in COBOL

Basic Procedure Division Commands

Selection Constructs

Iteration Constructs

Introduction to Sequential files

To top of page

Organization and Access

Introduction

Two important characteristics of files are *Data Organization* and *Method of Access*.

Data organization, refers to the way the records of the file are organized on the backing storage device. COBOL recognizes three main file organizations;

- Sequential - Records organized serially.
- Relative - Relative record number based organization.
- Indexed - Index based organization.

Method of access, refers to the way in which records are accessed. Some organizations are more versatile than others. A file with an organization of Indexed or Relative may still have its records accessed sequentially; but records in a file with an organization of Sequential, cannot be accessed directly.

Sequential Organization

Sequential organization is simplest file organization in COBOL.

In a Sequential file the records are arranged serially, one after another, like cards in a dealing shoe. The only way to access records in a Sequential file, is serially. A program must start at the first record and read all the succeeding records until the required record is found or until the end of the file is reached.

Sequential files may be *Ordered* or *Unordered* (these should be called Serial files). The ordering of the records in a file has a significant impact on the way in which it is processed and the processing that can be done on it.

Ordered and Unordered files

In an ordered file, the records are sequenced on some field in the record (like StudentId or StudentName etc). In an unordered file, the records are not in any particular order.

Ordered File	Unordered File
RecordA	RecordM
RecordB	RecordH
RecordG	RecordB
RecordH	RecordN
RecordK	RecordA
RecordM	RecordK
RecordN	RecordG

To top of page

Processing unordered Sequential Files

Introduction

It is easy to add records to an unordered Sequential file because we can simply add them to the end of the file by opening the file for EXTEND (e.g. OPEN EXTEND UnorderedFile). But it is not really possible to delete records from an unordered Sequential file. And it is not feasible to update records in an unordered Sequential file.

Adding records to an unordered Sequential File

To add records to an unordered Sequential File the file must be opened for EXTEND. Opening a file for EXTEND, positions the Next Record Pointer (NRP) at the end of the file, so that when records are written to the file, they are appended to the end.

The animation below demonstrates how records are added to an unordered file. As in all the examples in this tutorial, the records to be added are contained in a transaction file. The transaction file records are *applied* to the unordered file and the result is an unordered file that has all the transaction file records appended to the end of it.


Animation

Problems with unordered Sequential files


Detail

Although it is true that in standard COBOL, Sequential files cannot be deleted or updated "in situ", many vendors, including Microfocus, allow this for disk based files.

While it is easy to add records to an unordered Sequential file it is not really feasible to delete or update records in an unordered Sequential file.

Records in a Sequential file can not be deleted or updated "in situ". The only way to **delete** records from a Sequential file, is to create a new file which does not contain them; and the only way to **update** records in a Sequential file, is to create a new file which contains the updated records. Because both these operations rely on record matching, they do not work for unordered Sequential files.

Record matching

Updates to Sequential files are normally done in batch mode. That is, all the updates are gathered together into a transaction file, and then applied to the target file in one go. The target file is often called the *master file.*

There are few problems if we are just adding records to the end of a file, but if we want to update or delete records, then there must be some way of identifying the record we want to update or delete. A "key field" is normally used to achieve this.

A key field, is a field in the record whose value is unique to that record. For instance, in a student record, the StudentId is normally used as the key field. Without a key field to identify the record we require, we cannot delete or update records.

When we apply delete or update transaction records to a master file, we compare the key field in the transaction record with that in the master file record. If there is a *match,* we know we have to update or delete the master file record.

If both files are ordered on the key field, then this *record matching* operation functions correctly, but if either the transaction or the master file is unordered, record matching cannot work.

Attempting to batch delete records in an unordered file.

Suppose we have an unordered transaction file and an unordered master file. The transaction file contains records which indicate which records in the master file we want to delete. To delete the records we need to create a new master file which does not contain the deleted records.

The animation below demonstrates what happens when we attempt to batch-delete records from an unordered Sequential file.

To top of page

Processing Ordered Sequential Files

Introduction

An ordered Sequential file, is a file ordered upon some key field. The ordering of the records in the file makes it possible to process an ordered file in ways that are not available to us with unordered files. While it is not really possible to apply batch updates or deletes to an unordered file these are possible with ordered files.

Inserting records into an ordered Sequential file.

To add records to an ordered file a major consideration is to preserve the ordering. This means that the record must be inserted into the file in the correct position. It can't just be added to the end of the file as it can with unordered files.

As with all changes to ordered Sequential files we can't just insert the records into the existing file. To insert records into the file we have to create a new file that contains the inserted records.

The animation below demonstrates how records are inserted into an ordered Sequential file.

Self Assessment questions

The main processing loop from the animation is shown below. Examine the code in this program fragment and then attempt to answer the questions below.

Program Fragment

```
PERFORM UNTIL EndTF AND EndMF
  IF TFKey < MFKey
     MOVE TFRec TO NFRec
     WRITE NFRec
     READ TF etc.
   ELSE
     IF TFKey > MFKey
       MOVE MFRec TO NFRec
       WRITE NFRec
       READ MF etc.
     END-IF
  END-IF
END-PERFORM
```

SAQ

Q1	In the program, no allowance has been made for when the key fields are equal. What would it mean if the keys were equal?	Click the arrow for the answer
Q2	Why do you think the complex condition *EndTF **AND** EndMF* has been used to terminate the loop?	Click the arrow for the answer
Q3	There does not seem to be any special code to write out the remaining records when one file ends before the other. Can the code above be correct?	Click the arrow for the answer

---

Deleting records from an ordered Sequential file.

The animation below demonstrates how to delete records from an ordered Sequential file.

Animation

Updating records in an ordered Sequential file.

Updating a record requires a change to one or more of the fields in the record. Updates to Sequential files are usually collected together and applied to the file in one go - a batch. To update the records in an ordered Sequential file from an ordered batch of transaction records we have to create a new file that contains the updated records.

The animation below demonstrates how to batch-update records in an ordered Sequential files.

Multiple record type transaction files

In all the examples in this tutorial the transaction file has contained only records of one type or another. For instance, the transaction file contained either a batch of deletions, or a batch of insertions or a batch of updates. In real life though all these different kinds of transaction record would be gather together into one transaction file.

This raises some interesting problems. For one thing - the record sizes will be different. A deletion record only needs the key field, while an insertion requires the whole record, and an update may be somewhere between these two depending on how many fields we are updating in one go. There may even be different kinds of update record.

In future tutorials we will see how we can define and process files that contain multiple record types.

To top of page

Last updated : March 1999
e-mail : CSISwebeditor@ul.ie