Introduction to Minicomputers

File Organization

digital

1st Printing, June 1976
2nd Printing (Rev), October 1977
3rd Printing, August 1979

Printed in U.S.A.

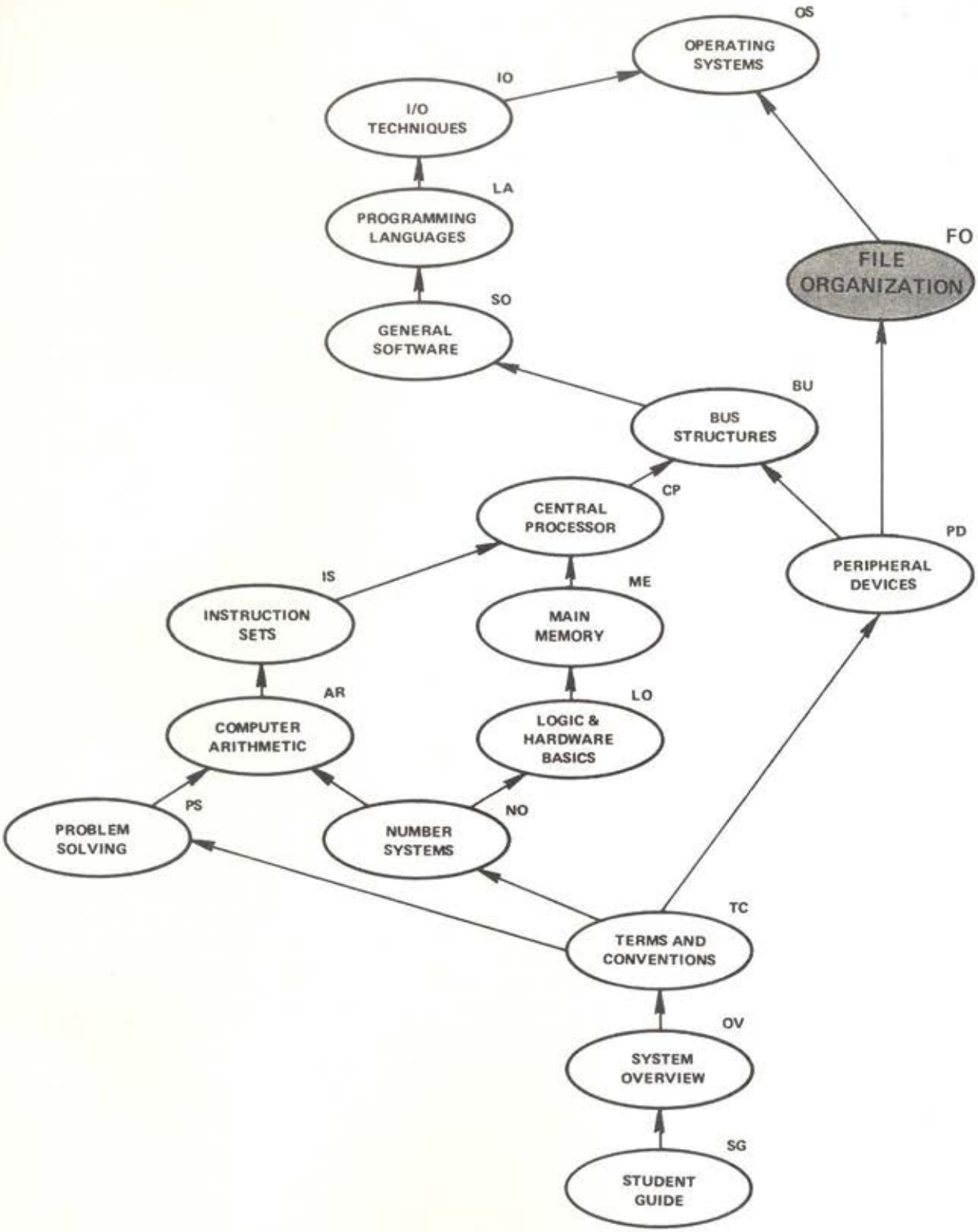
INTRODUCTION TO MINICOMPUTERS

File Organization

Student Workbook

Audio-Visual Course by Digital Equipment Corporation

COURSE MAP


OS
OPERATING SYSTEMS
IO
I/O TECHNIQUES
LA
PROGRAMMING LANGUAGES
SO
GENERAL SOFTWARE
FO
FILE ORGANIZATION
BU
BUS STRUCTURES
CP
CENTRAL PROCESSOR
PD
PERIPHERAL DEVICES
IS
INSTRUCTION SETS
ME
MAIN MEMORY
AR
COMPUTER ARITHMETIC
LO
LOGIC & HARDWARE BASICS
PS
PROBLEM SOLVING
NO
NUMBER SYSTEMS
TC
TERMS AND CONVENTIONS
OV
SYSTEM OVERVIEW
SG
STUDENT GUIDE

CONTENTS

File Organization

Introduction

Many computer applications require that data be stored during and between program executions. Commercial applications are good examples. Employee records must be *maintained between* paydays and *used on* paydays. Similarly, accounts receivable information about customers must be *updated* frequently although the data might only be *used* when account statements are to be produced.

Data is collected into *files* that are stored on auxiliary storage media such as magnetic tapes or disks. The organization of a file often determines how a file may be used. For this reason, file organization is chosen with the application's requirements in mind.

Lesson 1 of this module discusses the basic data structures that comprise a file and shows how these components are related to each other.

In the second lesson, you will learn about three types of files and how they are used in typical commercial applications. This lesson confines itself to a discussion of *uses* of these files. How files can be *organized* is discussed in the last lesson.

The third lesson is divided into three parts and discusses the four major types of file organization: sequential, direct access, indexed sequential, and list structured. Each file organization technique is described in terms of the following criteria:

- How records are arranged and located within the file
- How updates to the file are performed
- The advantages and disadvantages of the technique

Typical applications of each organizational technique are also shown.

Basic Data Structures

OBJECTIVE

Given data organization terms and definitions, be able to match each term with its definition.

SAMPLE TEST ITEM

Match each of these data organization terms with its definition.

Data Organization	Definition
Key Field	_____
File	_____
Record	_____
Field	_____
Character	_____
Library	_____

Definitions

1. A collection of units having one or more common characteristics or functions.
2. A single item of specific information.
3. A collection of files stored on magnetic tapes or disks.
4. Used for identifying and locating any part of a file.
5. Unit of usable data represented within the computer as a combination of bits.
6. A group of related items that are treated as a unit.

Mark your place in this workbook and view Lesson 1 in the A/V program, "File Organization."

Data is organized into several levels of complexity. This hierarchy of organization is the subject of Lesson 1. There are four levels in this hierarchy:

- Character
- Field
- Record
- File

Character

A character is the smallest unit of usable data – it is a single letter, numeral, punctuation mark, or other symbol (such as $ or %), and is represented within the computer as a unique combination of bits. Typically, a character code consists of seven or eight bits.

Field

Characters are seldom processed individually. Rather, several characters are processed as a single item of information. Just as single letters are combined to form words, characters are combined to form fields. A field is a single item of information (such as a name) composed of a sequence of characters. Fields contain specific kinds of information, such as employee names, social security numbers, addresses, etc. Because the amount of the data to be contained in a field can vary greatly, two different methods are available to store information in fields.

First, all fields of a single type (such as employee name) may be set to a fixed length. Thus, the employee name field may be defined to be 30 characters long. This *fixed* length method wastes space when the name is less than 30 characters. The excess space is simply unused or left blank.

A second method is to use *variable* length fields. In this technique, a field is as long as it needs to be. The actual length is determined by:

- preceding the field by an actual count of the number of characters in the field, or
- ending the field with a special terminating symbol (such as $).

Variable length fields use space more efficiently, but they do require more processing time.

Record

A record is a group of related fields that are treated as a unit. A typical payroll record for an individual employee might include fields such as name, social security number, address, gross pay, taxes withheld, etc.

Records also exist in an office. File folders within a filing cabinet are one example of records. Each folder contains the same type of information about various employees, customers, and so forth.

File

Records are collected into files. A file is a *collection* of records having one or more common characteristics or functions. Thus, a *file* describes a class of items, while a *record* describes a particular member or element of that class. Thus, a file drawer full of personnel folders is an example of a file in an office. In a computer system, files are usually stored on some auxiliary storage media such as magnetic tapes or disks. A collection of files stored on magnetic tapes or disks is referred to as a *library.*

To summarize, a hierarchy of data includes character, field, record, and file (Figure 1).

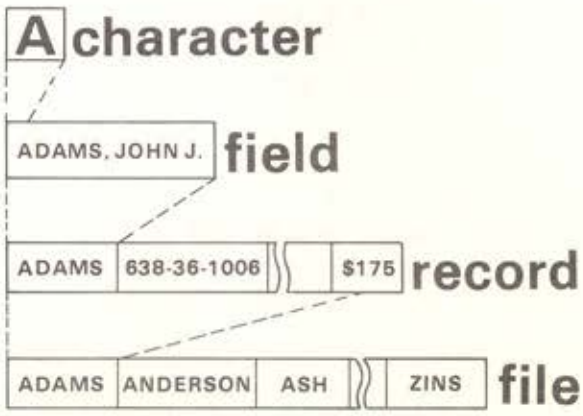


Figure 1 Data Hierarchy

Key Fields

A key field is a particular field within a record that is used to identify and locate individual records within a file. What distinguishes a key field from any other field is its use, and not how it is stored. A record may be part of more than one file if it contains more than one key field. For example, a particular record may be part of the personnel file (using the employee name as the key). At the same time, the record may also be part of the payroll file (using the social security number as the key). Thus, a key field may be either *alphabetic* (as a name) or *numeric* (as a social security number). The computer can locate any record in a file, provided that the correct key is given.

EXERCISE

Each of the following is an example of a term. Fill in the boxes with numbers corresponding to the terms.

☐ All the payroll data about John Doe

☐ The first letter in his name

☐ His gross pay

☐ All of the company's payroll data

☐ The data used for finding a record

(1) key

(2) character

(3) field

(4) record

(5) file

SOLUTION

Each of the following is an example of a term. Fill in the boxes with numbers corresponding to the terms.

[4] All the payroll data about John Doe

[2] The first letter in his name

[3] His gross pay

[5] All of the company's payroll data

[1] The data used for finding a record

(1) key

(2) character

(3) field

(4) record

(5) file

Types of Files

OBJECTIVES

1. Given statements of file uses and characteristics and three types of files, be able to match each statement with the type of file it refers to.

2. Given characteristics of types of records processing, be able to label those characteristics that refer to batch processing and those that refer to on-line processing.

SAMPLE TEST ITEMS

1. Indicate that each of these statements refers to a master (M), a transaction (T), or a report (R) file by writing the correct letter in the space provided.

Statement	File Type
Data extracted for special purpose.	_____
Relatively permanent information.	_____
Used for handling updates.	_____
.	.
.	.
.	.

SAMPLE TEST ITEMS

2. Indicate that each of these characteristics refers to batch (B) or on-line (O) processing by writing the correct letter in the space provided.

Characteristic	Type of Processing
Transactions are processed sequentially.	_____
More accurate updating of master file.	_____
•	•
•	•
•	•

Mark your place in this workbook and view Lesson 2 in the A/V program, "File Organization."

This lesson classifies files into three types by their uses: master files, transaction files, and report files. Each type will be discussed in turn.

Master Files

The master file is the complete set of records for a specific application. All records in a file have a common field (the key) by which they may be found. The records of the master file represent relatively permanent data such as names and account numbers. Some fields, however, are periodically updated, e.g., account balances, inventory on hand, etc.

Records are changed in the master file when:

- new records must be added to the file,
- obsolete records must be deleted from the file,
- occasional errors must be corrected, or
- records must be updated to reflect changes (in inventory, account balance, etc.).

However, from a practical standpoint, error correction and updating are essentially the same process.

Transaction Files

A transaction file is a set of records that indicate changes that are to be made in the master file. There are two important differences between master and transaction files. First, a master file has a record for *every item* in the class being stored (such as *all* active accounts or *every* inventory item). In contrast, a transaction file has a record for every change to be made. Many items in a class will not have records in the transaction file, but some items may have more than one record or transaction.

A second difference is in the contents of the records. Each record in the master file is likely to have many fields in addition to the key field. A transaction record is much simpler; it requires only three fields:

- a *key* to identify the master file record to be changed,
- an *operation field* to indicate what kind of change, and
- a *quantity field* to indicate the size of the change.

Hence, a transaction record is usually much shorter than a master file record.

Typical changes to a department store's billing master file might include:

- correct the customer's name
- change the billing address
- enter a new purchase (change the account balance)
- acknowledge receipt of a payment (change the account balance)
- open an account for a new customer (create a new master file record)
- close a customer's account (delete an old master file record).

Each change is specified in a transaction record along with the necessary data to perform the change (such as the new address or the amount of the new purchase).

Report File

A report file is a set of specific information extracted from a master file record to produce a report. Examples of report files are:

- a file of overdue accounts (from an accounts receivable master file),
- a file of out-of-stock parts (from an inventory master file), and
- a file of withheld taxes (from a payroll master file).

Report files are almost always generated on a periodic basis – whether it be daily, weekly, monthly, or at some other interval.

Using the Three Types of Files – On-line and Batch Processing

Let's see how the three files interact with each other and two methods of implementing these interactions.

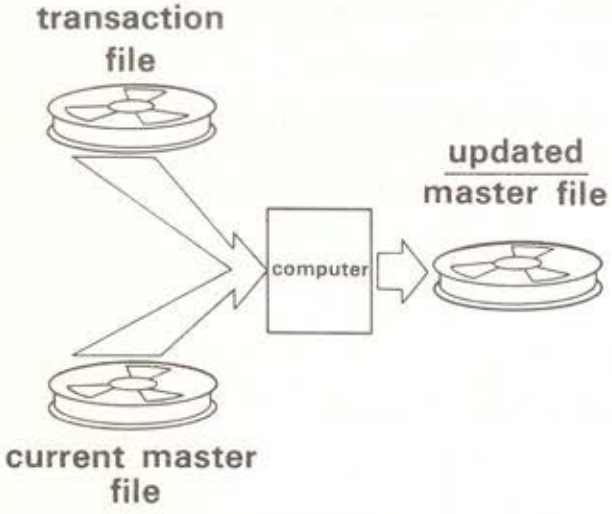


Figure 2 Updating the Master File

Figure 2 shows the interaction of the master file and the transaction file to produce an updated master file. Two different techniques can be used to implement this interaction.

On-line processing of transactions updates the master file as the transactions occur. Hence, the master file will always contain the latest information. This feature is particularly attractive for applications such as airline reservation systems where fast, direct access to the most up-to-date information is essential. When on-line processing is used, an actual transaction file is unnecessary, as each transaction is processed as soon as the computer receives it.

Batch processing of transactions requires a transaction file. As transactions are received by the computer, they are stored in a file. After some time, all of the transactions are processed, one after the other. Because the transactions are delayed, a file updated by batch processing is never as current as a file updated by on-line processing. On the other hand, batch processing can use simpler, more efficient, and less expensive file organization methods than can be effectively used with on-line processing.

Report files are usually generated *off-line.* Data is extracted from records in the master file and transferred into the report file for printing or display. Figure 3 shows how a report file is created.

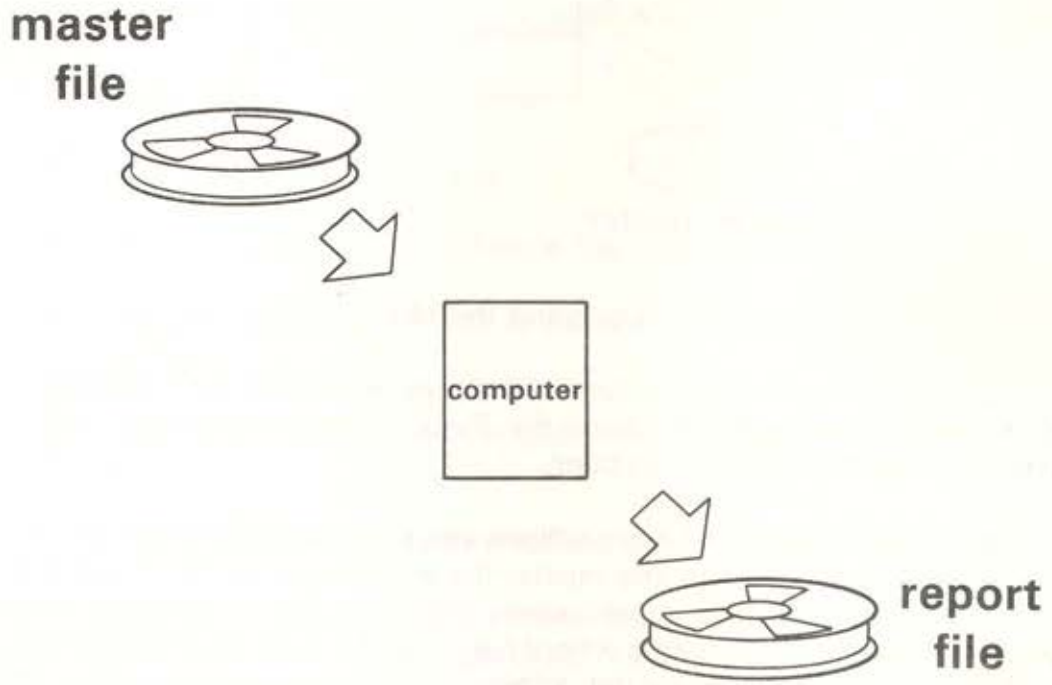


Figure 3 Creating a Report File

An example is the use of a report file to print a list of customers with overdue accounts. Each record is examined to determine if it meets the required condition (in this example, the requirement is that the account be overdue).

A report file produces a report of the current contents of the master file. Frequently, report files are generated on a periodic basis. As a result, they may become outdated quickly.

Figure 4 summarizes the three types of files you have studied.

TYPE OF FILE	CHARACTERISTICS
MASTER FILE	• relatively permanent records • reference information
TRANSACTIONS FILE	• transitory data • used for updating master file
REPORT FILE	• data extracted from master file

Figure 4 The Three Types of Files

EXERCISE

Fill in the blanks in the following sentences.

1. The electric company maintains a huge file containing information on all of its customers. Such a file is called a _____.

2. The company also collects a file of changes to this information which is used for periodic updating. The change file is called a _____.

3. Every month the company extracts from the first file some information on delinquent accounts. This information is used to print a list. The file of delinquent accounts is a _____.

4. The method of gathering changes and updating periodically is called _____.

5. An alternate method in which changes to a master file are applied immediately, is called _____.

SOLUTION

Fill in the blanks in the following sentences.

1. The electric company maintains a huge file containing information on all of its customers. Such a file is called a *Master file.*

2. The company also collects a file of changes to this information which is used for periodic updating. The change file is called a *Transaction file.*

3. Every month the company extracts from the first file some information on delinquent accounts. This information is used to print a list. The file of delinquent accounts is a *Report file.*

4. The method of gathering changes and updating periodically is called *Batch processing.*

5. An alternate method in which changes to a master file are applied immediately is called *On-line processing.*

Methods of File Organization

OBJECTIVES

1. Given a list of ten descriptions and characteristics of files and names of three methods of file organization, be able to match each statement with the method or methods it refers to.

2. Given seven descriptive statements, be able to label those that describe the pointer field in a list organization.

3. Given statements describing methods of addressing records, be able to label those statements that describe the hashing (or randomizing) method and those that describe the indexing method of addressing records.

4. Given a sample index file of eight key/address entries, be able to complete a binary search by: (1) selecting the entry being tested in each of three "tries," (2) indicating which entries have been eliminated after each "try," and (3) identifying the "try" in which the sought-after entry is found.

SAMPLE TEST ITEMS

1. Check the appropriate box or boxes next to each statement to indicate which file method or methods best answer the statement.

	Sequential	Direct Access	Indexed Sequential
a. Records are physically ordered by their key field.	☐	☐	☐
b. Records are physically stored in no particular order.	☐	☐	☐
.	.	.	.
.	.	.	.
.	.	.	.

SAMPLE TEST ITEMS

2. In the space provided, write a T if the statement correctly describes the pointer field in a list organization and an F if it does not.

Description	True or False
Depends in part on sequential storage of records.	_____
Contains storage location of the next record in the sequence.	_____
.	.
.	.
.	.

3. Indicate when the following statements describe the hashing (H) or the indexing (I) method of addressing records by writing the correct letter in the space provided.

Statement	Addressing Method
Address of a record is computed from the record's key field.	_____
Each record has an entry in a file table.	_____
.	.
.	.
.	.

SAMPLE TEST ITEMS

4. Perform a binary search by studying this sample index and then filling in the requested information.

Entry Number	Key	Address
1	110	167
2	157	671
3	286	1166
4	294	743
5	333	543
6	400	360
7	462	705
8	595	1023

Note: The search key is 333.

a. Write the entry *number* being tested in each try.

(1) Try one tests entry number _____ .

(2) Try two tests entry number _____ .

(3) Try three tests entry number _____ .

b. Indicate entries that are eliminated in each try by writing the entry numbers in the spaces provided. *Leave blanks if no entries are eliminated.*

(1) Try one eliminate entry number(s) ___ ___ ___ ___ .

(2) Try two eliminates entry number(s) ___ ___ ___ ___ .

(3) Try three eliminates entry number(s) ___ ___ ___ ___ .

c. The try that finds the correct entry is try number _____ .

Mark your place in this workbook and view Part 1 of Lesson 3 in the A/V program, "File Organization."

In the first two lessons we talked about the general properties and types of files. This lesson is divided into three parts and discusses how files are implemented in computers.

Sequential Files

Sequential files are the simplest and most commonly used file organization method. A sequential file is organized so that records within the file are physically arranged in ascending (or descending) order according to their key fields. Remember that key fields can be either alphabetic or numeric.

Sequential files are usually stored on magnetic tape because the limitations of the tape medium necessitate reading one record after another. Direct access to a specific record is not possible on magnetic tape. Sequential files can also be stored on magnetic disks.

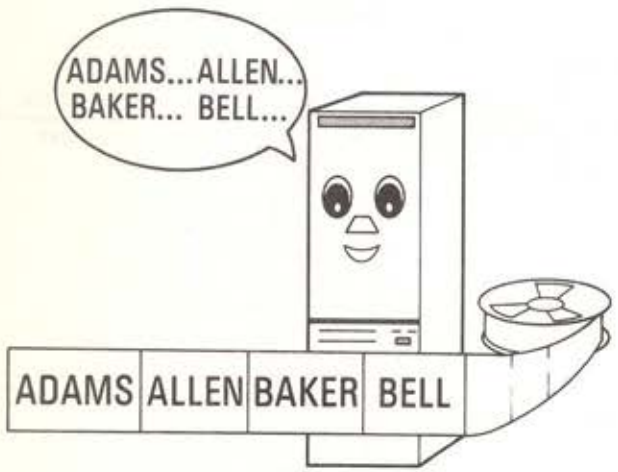


Figure 5 Locating a Record Sequentially

Records are located by the laborious process of starting at the beginning of the file and reading records one at a time until the desired record is found. Thus, searching for records near the end of the file is a very time-consuming process.

Updating of sequential files is normally done by batch processing to reduce the amount of time needed if records were to be processed individually. The transaction file is first *sorted* into the same order as the master file. Only one pass through the master file is required, regardless of how many transactions are to be performed.

This is how the updating process works. Keys are read, one at a time, from both the transaction and master files. As long as the transaction key is larger (assuming the files are in ascending order), the records in the old master file are simply copied into the new master file. When the keys match, the master file record is updated before being copied into the new master file. On occasion, the transaction key may be smaller than the master key. This situation means that a new record must be placed in front of the present master file record. Accordingly, data in the transaction record is used to create a new record that is immediately written into the new master file. Processing then continues with the next transaction. Figure 6 shows the flowchart for the updating process.

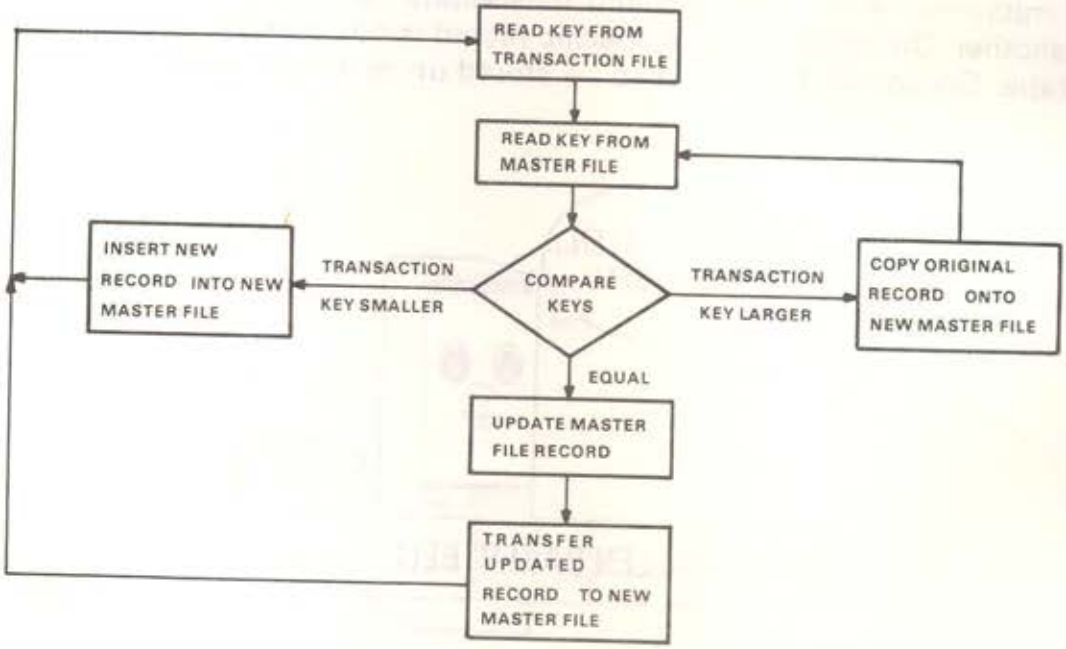


Figure 6 Updating a Sequential File

Advantages and Disadvantages

Sequential files are frequently used because they are the simplest method for organizing files. Either magnetic tapes or disks may be used to store sequential files. Because magnetic tape is an inexpensive storage medium, the costs of storing sequential files can be quite low. Sequential files are also more economical because they occupy the least amount of storage space. Since the physical arrangement of the records is the only structure in the file, no space is consumed for organizing the file.

On the other hand, sequential files have several disadvantages. Direct access to individual records is not possible. Hence, a significant portion of the master file must be read every time a single record is to be located. To achieve efficient updating of the master file, transactions must be sorted and then *batch processed.* Then, several transactions may be performed with only one pass through the master file. As a result, the master file is never completely up-to-date.

Finally, if the same file is to be accessed using more than one key, the records in the master file must be sorted again, each time a new key field is used. This can be a time-consuming and expensive process.

EXERCISES

1. How are records arranged in sequential files?

2. How does the computer locate a record in a sequential file?

3. List at least one advantage and one disadvantage associated with sequential files.

SOLUTIONS

1. Records are arranged in sequential files in ascending or descending order, based on a key field.

2. A computer locates a record in a sequential file by searching all preceding records until it finds one with the correct key.

3. Advantages of sequential files are:

   - simplicity
   - storage on inexpensive magnetic tape
   - use of less storage space

   Disadvantages of sequential files are:

   - computer must search to find record – can't go directly to it
   - transactions must be batched and sorted
   - to access files by more than one key, the file must be resorted.

Mark your place in this workbook and view Part 2 of Lesson 3 in the A/V program, "File Organization."

Direct Access

A *direct access* file is a file organized so that each record's location or address is determined independently of the other records and can be accessed without reading other records.

Direct access files are implemented on direct access (or random access) devices. Thus, direct access files are not found on sequential storage media such as magnetic tape. Rather, they are implemented on devices such as disks. Records on disks have unique physical addresses called storage addresses. Access to any particular location is accomplished *directly* if the storage address is known. Because of this ability to directly access any record in the file, individual records can be examined and updated quickly – without searching through the entire file.

Records are assigned to storage addresses in two ways: indexing and randomizing (also called hashing). Each method disperses a file's records within the disk area allocated to that file. Because the records are not stored in any particular order and because this arrangement appears somewhat "random" to the observer, the direct access method of organization is sometimes referred to as "random access organization." It must be remembered, however, that the records are not stored in a haphazard manner nor are they ever accessed randomly in an application. It simply means that the file has the capability for the records to be accessed in a random order.

"Randomizing" or Hashing

"Randomizing" or hashing (the preferred term) is the address calculation method in which a record's address is computed from the contents of that record's key field. For alphabetical keys, the letters of the key are usually converted into corresponding numbers, from 1 to 26. Next, a mathematical formula is used to combine all or several of the generated numbers into a single numerical value. When added to the starting address of the file, this value gives the storage address of the record.

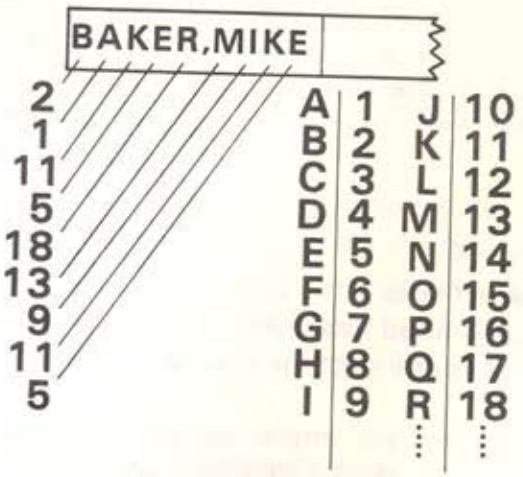


Figure 7 "Randomizing" or Hashing an Alphabetic Key

Notice that in Figure 7, the key BAKER, MIKE has been converted into the numbers 2, 1, 11, 5, 18, 13, 9, 11, and 5. One simple formula (or an unlimited number of formulae) might be to add all the numbers together and add the result to the file's starting address. In this case, the sum is 75. If the file's starting address were 2000, the storage address for the record with the key of BAKER, MIKE would be 2075. This process is applied consistently to all records in the file. *Ideally,* each record's key field converts into a *unique* storage address.

In practice, however, two problems occur. First, some storage addresses may never be used. Accordingly, some storage space is wasted. Secondly, more than one key field may convert into a particular storage address. In the A/V program we saw how BAKER, MIKE and CHASE, SARA could both transform into an address of 2075 if this simple formula were used. When two records contend for the same location, the usual procedure is to store one of the records in a separate "overflow area." The address on the overflow area is saved for later reference.

A third disadvantage of the hashing technique is that using a file with more than one key field is almost impossible. For instance, it is extremely unlikely that a hashing function or formula could be found that always generated the same address from a person's social security number as the address another hashing function produced from the person's name. Since there simply is no relation between a person's name and social security number, the two functions can never agree perfectly.

Despite these three problems, hashing techniques are frequently used because they are relatively easy to implement and are reasonably efficient while executing.

Indexing

Indexing is a method of determining a record's address by searching a table (or index) that equates record keys with their corresponding *storage addresses*. The table is stored as a sequential file in which each table record contains the *key* of a record in the data file and the *storage address* at which the record is located (see Figure 8).

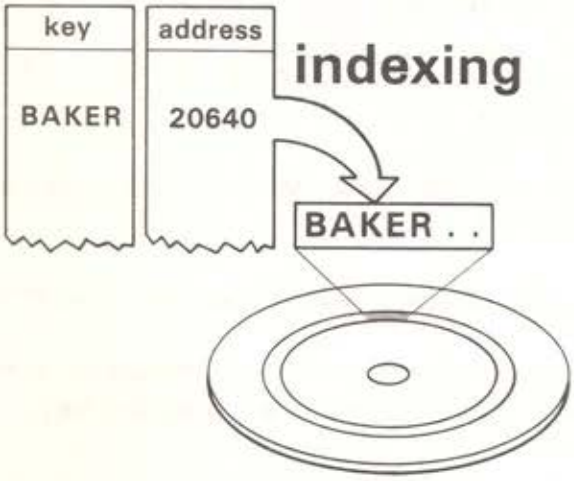


Figure 8 Indexing

There are drawbacks to this technique, however. First, the sequential nature of the index requires execution time to search through the index for the storage address. Secondly, the index requires storage space – a requirement that the hashing technique does not need.

On the other hand, files with multiple key fields are easily implemented by generating more than one index – one for each key field in the file. One table is used when searching by name; another is used when the social security number is the search key. Thus, indexing provides an ability that the hashing technique cannot.

Another advantage of indexing is that records may be placed in any location. Therefore, storage locations are usually allocated in numerical order as required. This means that although the order of the records is seemingly random, all records are part of a contiguous area of storage. No storage space is wasted in the file with unreferenced storage locations.

Further, the low efficiency of searching the sequential index can be improved by two techniques: binary searching and hierarchic indexing. *These techniques can also be applied effectively to sequential files in general to improve their performance.*

Binary Search

A binary search is a procedure in which a sequential file is searched by repeatedly halving the possible records until the correct one is found. The procedure can be stated as follows:

- Go to the middle record of all possible records.
- Compare the record key against the search key.
- If the record key is *higher,* ignore all records above and including this record.
- If the record key is *lower,* ignore all records below and including this record.
- If the record key is a *match,* the search is done.
- Repeat the procedure until a match has been found, or until all records have been eliminated (i.e., the key is *not* part of the file).

Figure 9 illustrates this procedure on an index file of sixteen entries (or records). Assume that the search key is 825.

1. The first try is made by comparing the search key (825) with the key of the eighth (the middle) record (value is 603). Since the entry key is lower, entries 1 through 8 can be eliminated.
2. The second try involves only records 9 through 16. Entry key 12 (the middle of 8 and 16) is compared against the search key. Because 950 (the entry value) is higher, index records 12 through 16 are eliminated from the search.
3. The third try involves only records 9, 10, and 11. Entry 10 is compared and found to be lower. Records 9 and 10 are eliminated, leaving only entry 11 still remaining.
4. Record 11 is a match with the search key, and address 2005 contains the complete data record for which we were searching.

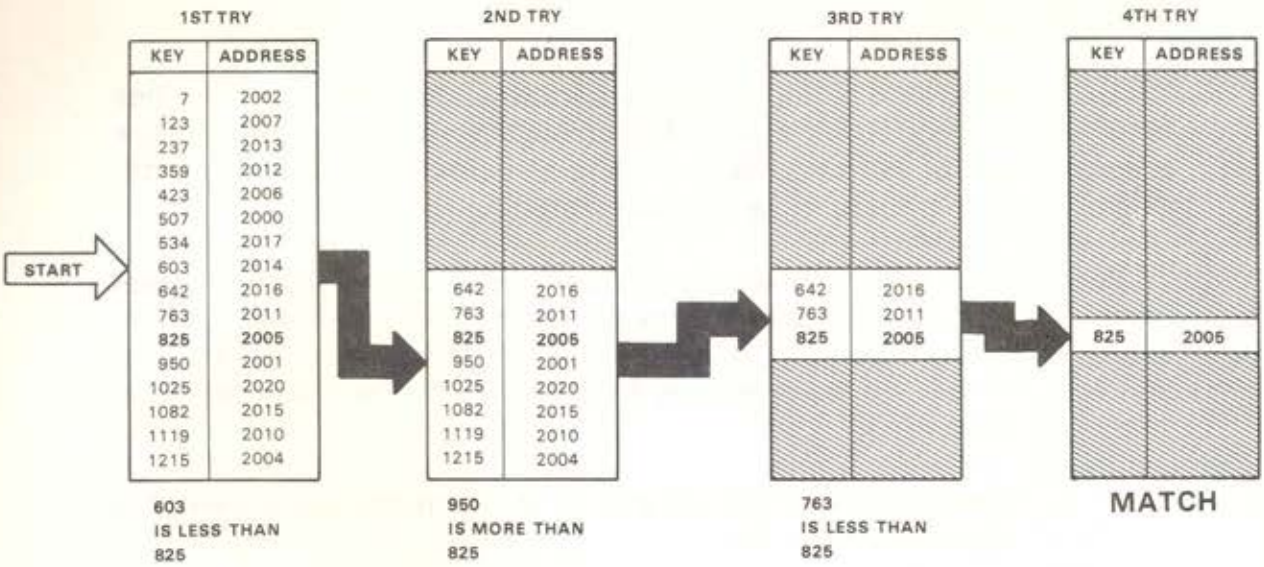


Figure 9 A Binary Search

One important point remains to be discussed. If index record 11 had NOT been a match, the search would have failed because the search key was not in the file. Only four comparisons were needed to determine definitely whether or not the key was in the file. Secondly, the example represents the worst case.

An important principle of binary searches is that *if there are 2^n records in a sequential file, the maximum number of comparisons needed for a search is only n.* Thus, 16 is 2^4, and only four comparisons are required. For a file of 1024 records, a maximum of 10 comparisons would be required. Notice that these are worst case estimates! A match could be found before the maximum number of tries is performed.

The primary advantage of a binary search is that it substantially reduces the search time in a sequential file. For very large files, the ratio of 2^n to n can be great.

This technique can be applied to sequential files in the general case, so long as the files are stored in a *direct access storage medium* such as disks. The jumping back and forth through the file is not practical on a sequential medium such as magnetic tapes.

Hierarchic Indexing

A second method of improving the performance of direct access files is the use of hierarchic indexing. Hierarchic indexing is the use of more than one level of indexing (i.e., providing an index to the index) in order to shorten the amount of sequential searching required.

Figure 10 shows an example of two-level indexing. The first level index consists of 26 records corresponding to the initial letters of the search keys. The second level index is then the actual index to the data file. Assume that we are searching for BARTLETT. The following steps would be performed:

1. Since BARTLETT starts with a "B," go to the second entry in the initial letter file.
2. The corresponding "address" (20) is the position in the main index of the first key beginning with a "B."
3. A sequential search is then begun at entry 20 in the complete index.
4. The record key BARTLETT is found at entry 22 and the storage address (2002) is used to access the data record.

Notice that four comparisons were required – one calculation in the initial letter index and three tries within the main index. A normal sequential search would have required 22 searches! Two-level indexing cannot equal the performance of a binary search in the worst case, because the records for an initial letter are still searched sequentially.

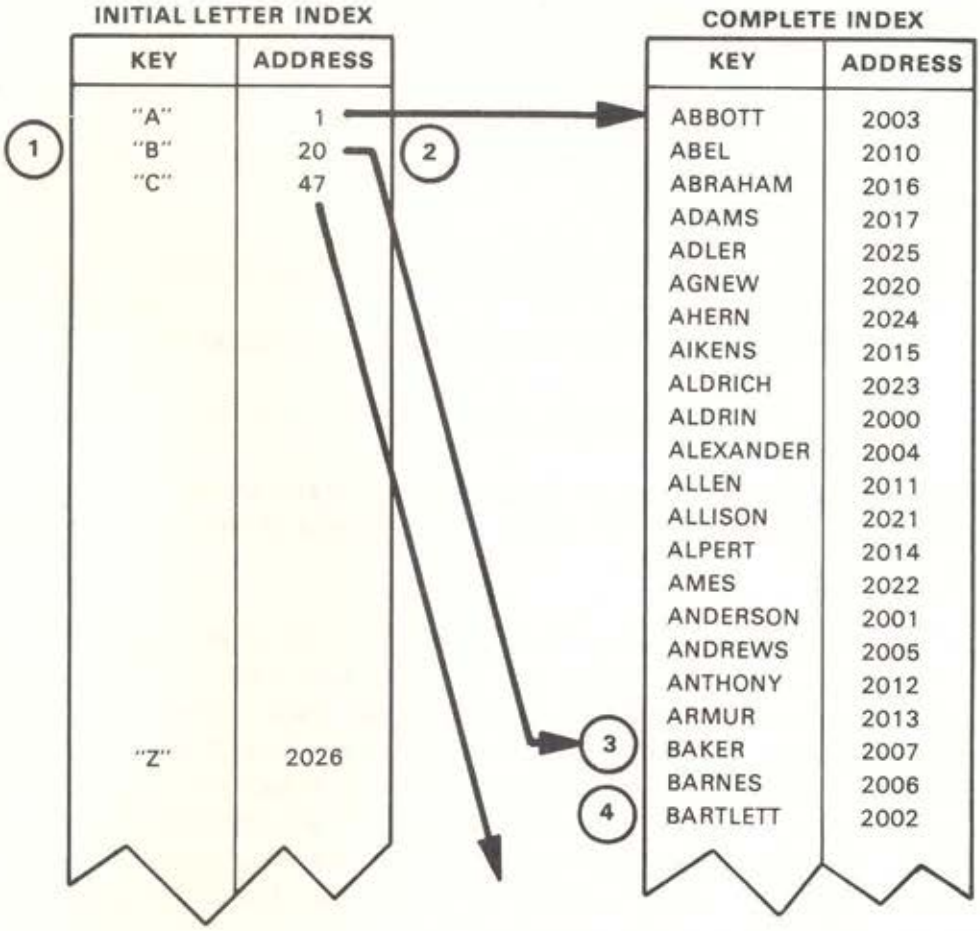


Figure 10 Hierarchic Indexing

More than two levels of indexing are possible. Indeed, some extreme systems have one level of indexing for every character in the key field. This method of distributing the key through many indices consumes a large amount of storage, but has one important feature. In the extreme case, the length of the search is more dependent on the length of the key (number of characters) than on the size of the data file. The distributed key technique is useful for very large files.

Another important point is that binary searching and indexing are *compatible*. That is, a file organization strategy can use both techniques to save searching tries in direct access files (or sequential files on direct access devices).

In summary, direct access files have two advantages:

- they provide direct access to any record, and
- files can be updated as transactions are received.

Thus, direct access files are very useful in applications demanding current data and fast search times. An example is an airline reservation system.

Direct access files also have disadvantages. Files cannot be stored on magnetic tape, but must use more expensive media such as disks. Secondly, direct access files require more space, either because of the gaps caused by hashing, or because space is needed for indices. Finally, direct access files are not really suited for applications involving most of the records in the file. Regardless of the addressing method used, direct access files require several operations for each access. This is far less efficient than sequentially processing an entire file in one pass – if most of the records are to be accessed.

You may wish to review this material before doing the Exercises on Page 41.

EXERCISES

4. How is the hashing method used to determine the location of a record on a disk?

5. Fill in the blanks in the following sentences:

   a. When the hashing method generates duplicate addresses, records are stored in the _____.

   b. The technique of associating keys with random addresses by means of a table is called _____.

   c. The technique of searching a sequential file by breaking it down into successively smaller halves is called _____.

   d. A structure for locating records with indices of indices is called _____.

6. List at least one advantage and at least one disadvantage of the direct access method.

4. How is the hashing method used to determine the location of a record on a disk?

   The hashing method is used to translate the key field of the record into a disk address.

5. Fill in the blanks in the following sentences:

   a. When the hashing method generates duplicate addresses, records are stored in the *Overflow area*.

   b. The technique of associating keys with random addresses by means of a table is called *Indexing*.

   c. The technique of searching a sequential file by breaking it down into successively smaller halves is called *Binary Search*.

   d. A structure for locating records with indices of indices is called *Hierarchy of Indices*.

6. Advantages of the direct access method:

   - Individual records can be examined or updated without having to search through intervening records in the file.
   - Transactions can be applied immediately, without batching and sorting.

   Disadvantages of the direct access method:

   - Direct access files cannot be stored on magnetic tape.
   - Efficiency is reduced when processing involves most, or all, of the records in a file.
   - Direct access files consume more space than do sequential files.

Mark your place in this workbook and view Part 3 of Lesson 3 in the A/V program, "File Organization."

Indexed Sequential Files

An indexed sequential file is a file stored sequentially on a direct access device with an index to provide direct access to any record. Indexed sequential files combine the features of both sequential and direct access files.

As might be expected, indexed sequential files are very flexible in operation. Multiple indices can be created so that several different key fields can be used for direct accessing of the file records. Additionally, applications using most of the file records can access the file sequentially, thereby saving processing time. Thus, an indexed sequential file can be accessed by the method that is most efficient for a specific application.

A major disadvantage of indexed sequential files is that they are not very flexible for adding and deleting records in the file. When a record is added to an existing file, the record is placed in a separate overflow area (see Figure 11). Two pointers are then established between the main file and the new record to insert the record in the main sequence of the file. Deleting a record is a straightforward process, but an empty space is created in the file where the deleted record had been located.

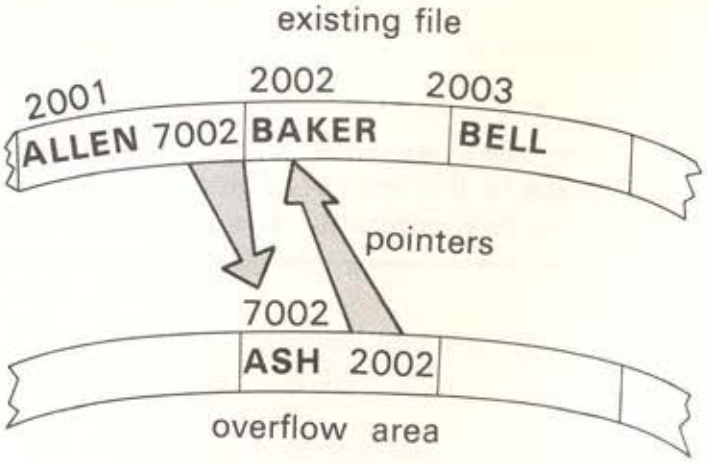


Figure 11 Adding A Record

Therefore, if numerous additions and deletions are made, frequent reorganization of the file may be necessary to reduce the wasted spaces in the file and to physically place the records from the overflow area into the main file area.

In addition, space is required for one or more indices. Thus, indexed sequential files have the maintenance problems of sequential files and the storage consumption of direct access files.

In summary, indexed sequential files have one major advantage – they may be accessed by either sequential or direct access methods, as applications require. On the other hand, there are disadvantages of the indexed sequential method:

- frequent reorganization of the file may be necessary;
- storage space is wasted by deleted records;
- overflow area must be maintained, and extra space is consumed by the indices; and
- more sophisticated software is required because of the more complex file structure.

List Structured Files

A list structured file is a file in which pointers or links are used to indicate the sequential order of the records. The records do *not* have to be stored sequentially.

Figure 12 shows a portion of a list structured file. Notice that each record has a pointer field that points to the *address* of the next record in the sequence. The last record of the file would have a special terminating pointer to signal the end. If a file is to be accessed by more than one key, a list of pointers must be maintained for each key. Thus, each record will have more than one pointer field.

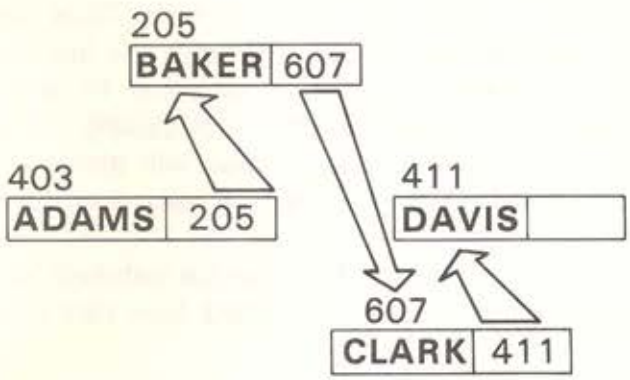


Figure 12 List Structured File

List structures have two advantages. First, records can be processed sequentially even though the records are not stored sequentially. Secondly, records can be added and deleted from the list very easily. To add a record, the following steps are performed:

1. An unused record address is located.

2. The record that immediately precedes the new record is located.

3. The pointer in the predecessor record is copied into the pointer field of the new record.

4. The pointer in the predecessor record is changed to contain the address of the new record.

The new record is now part of the list. Deletion of a record follows a similar procedure:

1. The addresses of the obsolete record and its predecessor are determined.

2. The pointer field in the obsolete record is copied into the pointer field of the predecessor record.

The obsolete record has now been deleted from the file's list. In many applications, the record would then be added to a "list of available records" for reuse at a later time.

List structured files have two disadvantages. First, the records must be larger to contain one or more pointer fields. Thus, the need for storage space may increase. Secondly, although the list structure behaves sequentially for accesses, it is not as efficient in terms of time as a simple sequential file. Because the processing of the many links requires extra execution time, an access will always take somewhat longer to be completed than if the file had been sequentially organized.

Notice that just as sequential files can be indexed to improve access times to individual records, list structured files can also be indexed.

EXERCISES

7. How does indexed sequential organization permit both sequential and random processing?

8. When is it necessary to reorganize an indexed sequential file?

9. List one advantage and one disadvantage of indexed sequential files.

7. How does indexed sequential organization permit both sequential and random processing?

   The file is arranged sequentially, allowing sequential processing. The index of record keys and disk locations allows random processing.

8. When is it necessary to reorganize an indexed sequential file?

   When a large number of additions and deletions have disrupted the file, with the result that the computer has to chase down a long trail of pointers to find a record, and efficiency is reduced.

9. List one advantage and one disadvantage of indexed sequential files.

   Advantage of indexed sequential files

   - It enables the user to choose either random or sequential processing.

   Disadvantage of indexed sequential files

   - Frequent reorganization is needed.
   - Storage space is required for the index and pointers.
   - Sophisticated software is necessary to make it work.

EXERCISES

10. Describe what takes place when a record is added to a file in list organization.

11. List one advantage and one disadvantage of list organization.

10. Describe what takes place when a record is added to a file in list organization.

    The new record is placed at the end of the file, and a pointer is appended to it, containing the location of the next record in sequence. Then, the previous record in sequence is located, and its pointer is adjusted to point to the location of the new record.

11. List one advantage and one disadvantage of list organization.

    Advantages of list organization:

    - Easy to add and delete records.
    - No indices involved.

    Disadvantages of list organization:

    - Pointer fields take up storage space.
    - Records cannot be processed randomly.

Summary

Before taking the module test, you may wish to review file organization. This chart summarizes the major areas covered in this module.

Criterion	Sequential	Direct Access	Indexed Sequential	List Structured
1. Arrangement of records within file	sequential	arbitrary	sequential	arbitrary
2. Normal method of accessing a record	sequential search through file	direct access through formula or table	direct or sequential methods	sequential search with pointers
3. Updates are handled by	reorganizing file	inserting record	storage in overflow area, eventual reorganization	manipulation of pointers
4. Advantages	• simple • can be stored on inexpensive mag tape • consumes little storage space	• fast access to any record • fast and easy updates are possible on-line	• may be accessed either sequentially or directly	• sequential access even though not stored sequentially • additions and deletions are easily performed
5. Disadvantages	• searching sequentially takes a long time • transactions must be batched and sorted • file must be resorted if a different key field is to be used	• magnetic tape can't be used • requires more space then sequential files • not suited for applications accessing most of the records	• frequent reorganizations are required • storage space is consumed by deleted records, overflow area, and the indices • sophisticated software required	• records are larger to hold the pointers • access time is somewhat slower then sequential because of the links

Take the test for this module and evaluate your answers before studying another module.