FILE ORGANIZATION OF LIBRARY RECORDS

I. A. WARHEIT: International Business Machines Corporation,
San Jose, California

Library records and their utilization are described and the various types of file organization available are examined. The serial file with a series of inverted indexes is preferred to the simple serial file or a threaded list file. It is shown how various records should be stored, according to their utilization, in the available storage devices in order to achieve optimum cost-performance.

One of the problems data processing people are beginning to face is the organization of library files. These are some of the largest and most voluminous files that will have to be organized, maintained and searched. They range in size from the National Union Catalog of the Library of Congress, which has over sixteen million records with an average of three hundred characters each, down to the hundreds of small college catalogs of 100,000 records. There are more than fifty universities whose holdings range from one million to over eight million volumes. The average holdings of library systems serving cities of 500,000 or more exceed two million volumes, although the actual number of titles is less. Since the turn of the century the university libraries have been growing exponentially and at present are doubling, on the average, every fifteen years.

Also the abstracting-indexing services, whose records are very similar to library catalog records and are used in much the same way, have grown very large. *Chemical Abstracts* which has been operating since 1907, now has over three and a half million citations. It provides data on some three million compounds and is today adding over a quarter of a million citations each year. If the present rate of growth continues, it will be adding 400,000 citations a year by 1971. *Index Medicus* and *Biological Abstracts* are very similar and there are a number of other somewhat smaller bibliographic services in the field of metals, engineering, physics, petroleum, urban renewal, atomic energy, meteorology, geology, aerospace, and so on.

In addition, library-type file maintenance, organization and search are being applied to medical records, adverse drug reaction reports, intelligence files, engineering drawings, museum catalogs and the like, and these too, represent very large information retrieval files. In other words, library files are very widespread and are beginning to become a problem for data processing.

CHARACTERISTICS OF FILES

The aforementioned library files have certain common characteristics. First, as already noted, they are large. In the next ten or fifteen years there will probably be several hundred libraries with holdings exceeding one million volumes each. Second, the records themselves are alphabetic and tend to be voluminous. They range from two hundred characters in an index journal, to three hundred characters for the standard catalog card up to two thousand characters for the abstract journals. In 1962 the Library of Congress, for example, estimated that it would need a file exceeding 9×10^8 bits to do its normal library processing and to store the serial records; it would need a file of 1.3×10^9 bits to store the circulation records and location directory and monitor the use of the collection, and would need a file of 10^{12} bits for the central catalog and the catalog authority files (1). On the basis of library experience since 1962, these figures are generally considered too low. Third, file records are variable in length. The librarian cannot control his inputs. The world's publications appear in every shape, form and identity and they must be recorded the way they have appeared so that they can be properly identified. Artificial identification such as book numbers, call numbers, Coden numbers for journals and the like are simply parochial conveniences and do not replace the actual bibliographic record.

Records in a large catalog file are generally stable and not dynamic. If there is a new edition of a document, a new bibliographic record is made. If the old document is retained along with the new edition, the old catalog record is also retained. The record is discarded only if the document is discarded and, in the large research library, this occurs very infrequently. New indexing or cataloging is seldom applied to old records. In contrast, the smaller item record file used for acquisition and processing, the circulation file, and the serials records file, all ranging from 10,000 to 100,000 records, are dynamic records requiring many and frequent changes, additions and deletions.

Each record item must have a number of different access points, since a single class or access point which everyone will accept is an impossibility. At present, with conventional library cataloging, card catalogs and printed indexes provide about five or six access points or records per title. However, computer systems, with their greater opportunity to do deeper indexing, are providing from ten to twenty keys or access points per title. Distribution of index terms is very uneven and not predictable. A few terms have a great many postings or addresses, while many terms, notably author entries, have only one or two postings.

File segmentation by subject class has been proposed by some data processing personnel, but inter-disciplinary needs are such that subject segmentation is not considered very seriously. File segmentation by date, especially for the abstract services, is increasing in popularity. It is generally thought that major activity, in the technologies especially, is concentrated in current records; this is less true, however, in the sciences and even less in the humanities. Public library and undergraduate library personnel may not object to segmenting their files, but those librarians responsible for major research collections that cover all disciplines do not look with favor on segmented files.

Although circulation records do provide some clues as to the activity of the various parts of a library's collection, no one really knows what the search activity in the catalog is, or how it is distributed across the various records used. Therefore, since every record is considered permanent in libraries, major effort has been expended on input processing which has included the recording of much material whose utility is questionable.

A user wants to access files in open language, and wants to receive response in open language; he will not use codes and so-called machine language and will tolerate only a minimum of training on methods to interrogate the file. He prefers to engage in an actual dialogue with the file and if he cannot do this will ask a reference librarian or reader's advisor to find the references for him. He also wants real-time response. If he doesn't get fairly prompt answers, he will go elsewhere to satisfy his informational needs.

TYPES OF FILES

The librarian must work with a number of files: 1) The item record file is the record of an item, book, journal, report. etc., that is being ordered, is on order, is being received, or is being processed by the cataloger. 2) The catalog file is the permanent bibliographic and subject record of the item that has been processed by the cataloger. 3) The serials record file, which is in two parts, is the record of holdings of completed volumes both bound and unbound, and the check-in record of currently received periodical issues. 4) The circulation control file keeps the record of all items loaned or otherwise charged out. 5) The catalog authority file is the thesaurus-like vocabulary control which indexers and catalogers use as their authority list and guide in assigning index terms. It is also used to "normalize" the inquiries of a searcher and convert them to legitimate index terms.

The librarian is also concerned with a number of indexed abstracts produced by various discipline oriented institutions which are used in libraries. He also uses a number of special files: borrower or patron file, special collection files, location files, vendor files, and the like.

Except for a few comments about the item record, this discussion is confined primarily to the catalog file, which is by far the largest file and, for the librarian and the general user, the most important. As already noted, in most respects it is very similar to the indexed abstract file and, in fact, in certain special libraries, these two files are combined.

IN PROCESS FILE

The in process, or item record, file consists of records of all items which the library is acquiring and processing. It is not a very large file, or, at least if properly policed, should not be. Unfortunately, because in manual systems it is difficult continuously to follow up outstanding orders, a lot of deadwood accumulates and files become unnaturally large and difficult to handle. In a well controlled file, however, the number of records does not grow appreciably, for, although new items are added, processed titles are removed when they are added to the catalog file.

In addition to providing such normal bibliographic access points as personal author, corporate author, title, report number and the like, the item record may also be searched by a number of specialized keys: order number, vendor, publisher, journal code, contract number, fund, requester.

The item record is very dynamic. Information available to the librarian when the order for an item is placed may be faulty. New information will be coming in about the item, such as price, shipping costs, invoice number, change in vendor, and change in title. Various funds have to be charged and obligations changed, payments authorized, funds decremented, receipt notices prepared and sent to requesters, flags in various files changed to prevent duplicate orders and the bibliographic record transmitted to the cataloging staff. However, once an item has been received and cataloged, only the bibliographic information (author, title, place, publisher, date, pagination) are retained and the rest of the information is retired to an historical file. (2).

Because it would provide greater flexibility as new and unexpected demands are generated, the best way to handle this dynamic file would be with a generalized data management system rather than with a tailor-made acquisitions and processing program. Although present data management systems are really not suitable, because of variable length records in item record files and because terminals will be used, it appears that some could be adapted.

CATALOG FILE

The tendency today, however, is to build a single master file with various functional fields where bibliographic information, ordering, and purchasing data, loan records, location information and other item control data are stored. How should this very large master catalog file be organized so that it will be easy and economical to maintain and provide all the desired search capabilities?

There are three basic file organization schemes in use today for information retrieval: the serial file, the inverted file and the list process file (3,4,5). Actually, from a technical point of view, both the inverted file and the list process file represent two different classes of list structures and are, therefore, sometimes referred to as the inverted list system and the threaded list system.

Serial File Organization

Although the serial file is the easiest and cheapest to maintain, the librarian obviously cannot accept purely serial searching of his catalog. The file is much too big and the real time requirements are such as to rule out any but the shortest, simplest serial or sequential search. As will be pointed out later, the librarian does need some serial searching capability, and of course he does need it if he wants to do any browsing. However, if he is to provide any kind of useful service, he must use direct-access storage devices and access to his records individually.

Threaded List File Organization

For a while there was some interest in using a threaded list file organization for the catalog file. Here, the searcher is first directed through a dictionary or directory to the latest record associated with a term. This record also contains the chain address of the previous record having the same descriptor, so that a user can run through a "chain" or "list" until he reaches the oldest or last record, or comes back full circle to the starting record. Each record belongs to a number of lists, one for each descriptor used to describe it, and there are as many lists as there are descriptors.

Such a system seems economical of storage space in that a secondary or separate index does not have to be stored, but, since storage space for the chain or link address has to be provided, the actual savings are very small. There are several possible refinements of this list file organization which reduce storage costs. Some involve elimination of redundant information; a term, or any other searchable piece of information, is stored just once, sometimes in the form of a table. Each record that contains searchable information has a pointer to the term itself. There have to be, of course, pointers from every term back to the records as well. Insofar as the pointers may require fewer bits than the terms or addresses themselves, there is a saving in storage space. It does cost some additional processing time and file maintenance is somewhat complicated. (6).

Another economy measure is provided by what is generally called a multilist system which groups several—usually three—descriptors into one super key with one chain address. A multilist not only saves space but also speeds both file posting and searching by processing multiple descriptors simultaneously. (7,8,9,10,11). Such a system, to be workable, must permit grouping of various descriptors into mutually exclusive groups, and within each group there must be some equitable distribution of descriptors posted to records. In normal library information retrieval applications, a very large percentage of the descriptors are used just for one or two documents and only a few descriptors are used to identify a large number of document records. In other words, most of the so-called super keys end up having just a single real descriptor, which is equivalent to establishing a separate list for each descriptor. In a test made with the Defense Document Center collection it turned out that about ninety percent of the super keys had only single descriptors. (12,13).

There are, in addition, special modifications of multilist files which essentially involve segmenting the multilist to fit the hardware, for example, the track length or cylinder size. (14). A fragmented sub-list, sometimes referred to as a cellular multilist, may even contain all the link addresses in the directory, thus becoming indistinguishable from an inverted file.

Any list process file organization, however, does pose serious file maintenance problems, especially where individual records must be changed or deleted. Also special precautions must be taken to avoid broken chains and provision made to repair breaks, although some advocates of list process files claim it is easier to maintain threaded lists than inverted lists. Of course, if multilists are used, a special effort must be made to build the super keys.

It must not be forgotten that a threaded list directory can only provide the search statistics for a single term and, unlike the inverted list, can only provide intersection statistics upon completion of a total search. The few librarians who have been exposed to threaded list file organization have not reacted favorably. A few have been interested in applying this technique to do hierarchical searches and other relationship connections in their authority lists or thesauri, but have not seriously considered using it for their catalog files.

Inverted File Organization

The traditional library file organization as exemplified by the standard card catalog has been based on a serial main file plus an inverted file. Here a normal serial file is "inverted" and the file sequenced by index entry or key. The record itself is duplicated under each of its keys, which librarians call tracings. By strictly limiting the number of tracings or keys applied to each record, the librarian can keep the card catalog down to a reasonable size. However, as deeper indexing is applied to the documents, more keys or tracings are used and the file becomes very large.

Furthermore, storage costs in the mechanized file are appreciably higher than in an ordinary manual card file. The full record, therefore, in a mechanized system cannot be economically stored behind each term. Only the document or record number or file address of the master record is recorded after each term; in other words, the inverted file is just an index to the record file. The main record file itself is a simple serial file where each record is complete in itself, the tracings or keys in the record and the address of the record being duplicated on the inverted file. The catalog file, therefore, is made up of two parts: a serially organized main or master record file, and an inverted index to the main file. (15).

Maintenance of an inverted index is expensive. Tracings and the addresses to which they refer have to be duplicated, requiring costly additional storage space. New terms and new addresses cannot simply be added to the end of a file but must be distributed and interfiled throughout the index, causing a number of file maintenance problems. The inverted index and main serial file must be kept in phase, with changes in one being reflected in the other. To maintain these files, separate inputs should not be prepared; instead the inverted index should be generated from the main record file update by program control. (16,17,18,19).

Although the combined file organization of a serial record file and an inverted index does cost more to maintain than serial or list file organization, it provides such superior search capabilities that it has become the favored library catalog file organization.

Since the inverted file is organized by subject headings or descriptors and since a search request is specified by listing the desired descriptors and their logical relationships, the search programs need only examine the items filed behind each selected descriptor or subject heading. It is unnecessary to look at all the records, as it is with the serial file. The inverted file search, in its basic form, takes the request descriptors, obtains the list of record addresses or items under each relevant descriptor, makes the specified logical connections, and produces all items satisfying the request. The search procedure examines only potentially pertinent records, ignoring the rest of the file. In other words, the file is organized every time a search is made to suit the requirements of the search. Thus, the file and the request are compatible and utilization of the file is essentially independent of its size.

An inverted index provides a very special capability to a searcher who is using a terminal, on-line system. He can test both individually and collectively the effectiveness of the terms of his search statement without having to make a complete search of the master record, simply by examining the inverted index. The system will tell him, for example, the number of entries under a term. It will tell him how many entries several terms share in common so that he can test the intersections, that is, the conjunction and disjunction of the terms. The count of addresses that results from the list intersection can be returned immediately to the terminal as an upper limit of the number of hits. In effect decoding of the Boolean expression takes place in the inverted index, which is a very compact list, and hence the response time is fast. It is true, some additional calculations and comparisons in the record itself may reduce the number of hits, but will never increase them.

Sitting at a terminal, a searcher can ask the system what will be the maximum number of hits he will get in response to a search statement. He can change the parameters of his search statement and see immediately what effect that will have on the response of the system. It is primarily because of this capability of the user to have a dialog with the machine that every terminal-oriented library information retrieval system, at least of which the author is aware, is adopting an inverted file organization.

In order to reduce storage costs, not every search term need be carried on an inverted index. Those search terms or index entries that are practically never searched alone, but used rather in conjunction with another term or tracing, are carried only in the main file and not on the inverted index. In a library catalog these terms are usually the place and date of publication, publisher, language of the book, level of the publication (i.e. adult, children, youth), number and type of illustrations, and so on. These terms appear on almost every record and some of them are high density terms; that is, they are heavily posted. For example, in a typical U.S. library, some eighty per cent of the books are identified as being in English. Form headings (bibliography, essay, poem, biography, map, etc.), geographic headings, and numerics that are used in conjunction with what are called main headings, also do not appear on the inverted index, but can be searched in the main file. In the very unlikely event that a search is required to be made only for a term not on the inverted file, then, of course, a serial search can be made of the master file. In some systems, a very compact serial file of data may simplify serial searching of the master file.

PHYSICAL ORGANIZATION

A basic understanding of how a library's records are used is necessary to a proper plan for their physical organization. In a manual system, logical organization and physical organization of a library's records are identical. Furthermore, all files are physically the same, usually on 3x5 catalog cards or, in a few cases, in printed book or sheaf catalogs. In a computer system, however, because of varying capacities, speeds and storage costs of different direct access devices, it is extremely important that the various records and segments of records be stored in those devices which will give the best cost-performance for the application. This means that the rate of utilization of the various records and parts of records, as well as the size of the records, will determine what types of devices will be used as physical files.

In a library operation there is very heavy use of index terms, or subject headings and author entries, to search the files; records for these entries can be very short. Borrower records and charge-out records in circulation control systems are also very actively used. There is less use made of the bibliographic record or journal citation. These records are somewhat longer than the subject and author tracings, and hence require more storage, but do not need such rapid access. Notes, abstracts and other explanatory material can require an enormous amount of storage space but, as a rule, are used only infrequently. Patron registration, as contrasted with borrower records, is used much less frequently, unless, of course, the two types of records are combined. Since serials holdings records do not change very frequently, printouts are quite satisfactory as finding tools and the records are usually kept off-line. Journal check-in, however, requires a great number of accesses every day.

In view of the requirements generated by the above uses, the present thinking for on-line library systems, in terms of current hardware, runs something like this: In a combined file system described above, with the bibliographic record on the serially organized main file and the index in an inverted file arrangement, the inverted file, which must be accessed many more times than the main file, would best be carried on disk files. The bibliographic record itself, being much more voluminous and accessed less frequently, is stored in a larger, slower, more economical file like the IBM 2321, the Data Cell. Abstracts, and other seldom used bulk records might well be on tape, off line. Actually, though, as libraries build up their record files to control their total collections, they will, of course, exceed the capacity of the present Data Cells and will have to go to future mass memory devices similar to the IBM Photo Digital Storage System. Then it may be economical to put even abstracts and notes of the bibliographic record on line.

If there is a separate item record file of in process or acquisitions data, it can be handled in the same way as the catalog file, that is, all access points as an inverted file on disk with the record itself on the Data Cell strips. If, however, the total item record file is not too big, it might well be stored on disk. Circulation control records are carried on disk, but patron registration, if it is to be kept on line, would be more economically stored in the Data Cell.

The authority list or thesaurus really has two functions. It is heavily used to validate and convert all inputs and all search requests. It is also used to store all cataloging and indexing decisions and to provide guides to users as to the formulation of search queries. The necessary data makes for long records that are either infrequently used or available as printouts. Therefore, a condensed form of the authority list or thesaurus, a form which carries only the terms and their equivalents, is best stored on disk, whereas the full-blown authority list which is used primarily for printing the thesaurus and its supplements can be carried off-line on tape, or in the cheapest, biggest and slowest direct access device which is available.

In order to achieve economical, compact storage, the subject headings, descriptors or index terms would not be stored in open language but in numeric codes. By using, for example, the decimal code as used in a Dewey decimal system, numeric codes would also make it possible economically to build hierarchies or class tables with the descriptors. It would be necessary, therefore, in every transaction, to translate from open language to code when interrogating the system and to translate from code to open language when outputing from the system. Translations would have to be very fast to accommodate the traffic of a large number of terminals. The translation job, using a stored table, might have to be done in an auxiliary, large core storage, which is very fast but more expensive than disk files.

As a general rule, what is being proposed is that for very large files the index and the bibliographic record are not to be stored in the same device. One might start this way until the file and the traffic into it are built up and the system becomes fully operational. However, the system should be so structured that indexes could be stored in files that are faster than the bulk storage devices used for the records. The translation files, that is, the tables that convert from open language to stored codes on input and the reverse on output, can be stored in the fastest available exernal storage. (20).

It is extremely doubtful that hardware development in the immediate future will change these principles of library file organization very much. As storage costs drop, total capacities increase, and access times become shorter, more and more libraries will find it practical and economical to put their files on line in order to provide the improved services that users demand.

REFERENCES

1. U. S. Library of Congress: *Automation and the Library of Congress* (Washington, Government Printing Office, 1963), p. 74.
2. Batts, N. C.: "Data Analysis of Science Monograph Order/Cataloging Forms," *Special Libraries,* 57 (October, 1966), 583-586.
3. "Corporate Data File Design," *EDP Analyzer.* 4 (December, 1966).
4. Climenson, W. D.: "File Organization and Search Techniques," *Annual Review of Information Science and Technology.* 1 (New York: Interscience, 1966), p. 50.
5. Borko, H.: "Design of Information Systems and Services," *Annual Review of Information Science and Technology,* 2 (New York: Interscience, 1967), p. 50.
6. Castner, W. G., et al.: "The Mecca System - A Modified List Processing Application for Library Collections," *Proceedings - A. C. M. National Meeting* (1966), pp. 489-498.

7. Prywes, N. S., et al.: The *Multi-List System* (Philadelphia, Moore School of Electrical Engineering, University of Pennsylvania Technical Status Report No. 1 under Contract NOnr 551(40), November, 1961).
8. Prywes, N. S.; Gray, H. J.: "The Multi-List System for Real Time Storage and Retrieval," *IFIP Congress Proceedings.* 1962, pp. 112-116.
9. University of Pennsylvania, Moore School of Electrical Engineering: *The Tree as a Stratagem for Automatic Information Handling* (Report of Work under... Contract NOnr 551(40) and... AF 30(602)-2832, Moore School Report No. 63-15, 15 December 1962).
10. Lefkovitz, D.: *Automatic Stratification of Descriptors* (Philadelphia, Moore School of Electrical Engineering, University of Pennsylvania, Technical Report under Contract NOnr 551(40), Moore School Report No. 64-03, 15 September 1963).
11. Landauer, I.: "The Balanced Tree and its Utilization in Information Retrieval," *IEEE Transactions on Electric Computers* (December, 1963), pp. 863-871.
12. UNIVAC Division of Sperry Rand Corporation: *Multi-List Systems: Preliminary Report of a Study into Automatic Attribute Group Assignment;* Technical Status Report No. 1-2, 3#AD 609 709, 4#AD 609 710 (1963-1964).
13. UNIVAC Division of Sperry Rand Corporation: *Optimization and Standardization of Information Retrieval Language and Systems; Final Report* (AD 630-797, 1966).
14. Lefkovitz, D.: *File Structures for On-Line Systems* (class syllabus).
15. Curtice, R. M.: *Magnetic Tape and Disc File Organizations for Retrieval* (Lehigh University, Center for Information Sciences, July, 1966).
16. Warheit, I. A.: "The Direct Access Search System," *AFIPS Conference Proceedings,* 24 (1963), pp. 167-172.
17. Warheit, I. A.: *The Combined File Search System. A Case Study of System Design for Information Retrieval* (paper presented at the F. I. D. Meeting in Washington, D. C., October 15, 1965; Abstract, 1965 Congress, International Federation for Documentation (FID), Washington, D. C., U. S. A. 10-15, October 1965), p. 92.
18. Prentice, D. D.: *The Combined File Search System* (San Jose, California: IBM June 15, 1964).
19. *1401 Information Storage and Retrieval System - Version II; The Combined File Search System, No. 10.3.047* (Hawthorne, New York: IBM, May 1, 1966).
20. *Warheit, I. A.: File Organization for Libraries; Report to Project Intrex, MIT, Cambridge, Massachusetts, March 14, 1968.*