

P.R.L.
TECHNICAL NOTE

TN-87-58

COMPUTERIZED PERSONAL DOCUMENTATION SYSTEM

Neelam Mehta, R.R. Bharucha
and
G.G. Dholakia, D.R. Kulkarni

July, 1987

PHYSICAL RESEARCH LABORATORY
AHMEDABAD-9

DOCUMENT CONTROL AND DATA SHEET

1. REPORT NO. : PRL : TN-87-58

2. TITLE AND SUBTITLE : "Computerized Personal Documentation System"
3. REPORT DATE : July, 1987
4. TYPE OF REPORT : TECHNICAL NOTE
5. PAGES : 33
6. PRICE : Unpriced
7. NO. OF REFERENCES :

8. AUTHORS : Neelam Mehta, R.R. Bharucha, G.G. Dholakia & D.R. Kulkarni

9. a) PURPOSE : To record and organize personal documentation
b) USEFUL FOR : For immediate, precise and indepth information retrieval.

10. ORIGINATING UNITS/DIVISIONS AND ADDRESS :

Library
Physical Research Laboratory
Navrangpura
Ahmedabad 380 009

11. SPONSORING AGENCY : Nil

12. ABSTRACT : The computerized personal documentation system MYLIB enables professional persons to retrieve relevant and pertinent information from their personal collection.
A large part of man's reading is wasted and can never be found again unless there is a good indexing system. Since manual indexing is time consuming & complicated a mechanised indexing system is used.
This user friendly system enables even a novice user to retrieve the required information.

13. KEYWORDS : Computerised Information Retrieval, Personal Documentation, Mechanised Indexing

14. DISTRIBUTION STATEMENT : Distribution Unlimited

15. SECURITY CLASSIFICATION: Unclassified

C O N T E N T S

I.	INTRODUCTION	...	1
II.	WHAT IS MYLIB	...	6
III.	HOW TO CREATE PERDOC. DAT FILE	...	10
IV.	HOW TO LOAD THE INFORMATION	...	14
V.	HOW TO RETRIEVE INFORMATION	...	20
VI.	HOW TO UPDATE INFORMATION	...	23
VII.	CONCLUSION	...	25
VIII.	APPENDIX-A STRUCTURE OF THE SYSTEM	...	27
IX.	APPENDIX-B INSTRUCTIONS TO USE MYLIB	...	31
X.	APPENDIX-C LIST OF STOP WORDS	...	33
XI.	APPENDIX-D EXAMPLE OF AN INFORMATION RETRIEVAL SESSION	...	34
XII.	FIGURE	...	35

Computerized Personal Documentation System

Neelam Mehta, R.R. Bharucha

and

G.G. Dholakia, D.R. Kulkarni

ABSTRACT

Every professional person owns his personal collection of documents (books, letters, reports, reprints, journals, cuttings, photocopies, slides etc.) which need to be well organized and accessible at any time. A computerized personal documentation system MYLIB has been developed on the computer system DEC-1091. The system MYLIB enables professional persons to retrieve precise indepth information from their personal collection with the aid of mechanized indexing. Retrieval of documents is also possible through any field in the document record. The main strategy used in the system is to invert the document file on various fields required for accessing the document. It is generally experienced that such a system leads to better and more intensive utilization of the available information. This in the long run stimulates creative thinking and improves ones method of working. The system is quite user friendly and can be used even by the novice user without much difficulty. This report describes in detail the design and implementation of the system MYLIB.

Chapter 1

INTRODUCTION

Persons in every profession, be they scientists, engineers, managers etc. have their own personal collection of documents. These could be books, journals, photocopies, slides, letters, microforms, notes, reprints, references to material not in ones collection, etc.

Majority of people read much and remember very little. The information contained in the material is often received when it is not immediately needed and if not organized properly, it is lost in the large bulk of accumulated documents, and is almost impossible to trace when required urgently.

Personal documentation deals with the organisation of one's collection of documents and the technical methods used to store and retrieve the information.

Owing to the tremendous progress made in many fields, it becomes essential to improve ones method of working by using advanced techniques. Most people are averse to change and may feel that there is no need for a documentation system, but they forget that they use only a small part of the documents available with them. This happens because we tend to use only the information that we can easily find.

Owing to the multidisciplinary nature of science and technology we need more and more information. In the words of John Naisbitt - "we are drowning in information but starved for knowledge". Knowledge is after all a person's range of assimilated information.

We tend to spend almost half an hour to one hour to read an article or a report. If it contained some useful facts or ideas would it not be worthwhile to spend a few minutes to record them and to store the information so that it would be accessible when required? This small time spent would save hours or days of work later.

Even in an advanced country like the USA where hundreds of online databases are available which give access to information, it is estimated that over two billion dollars are wasted a year on duplicated activities in research and development.

A personal documentation system does not just save time and cost. The more important fact is by providing quick and easy access to information, it enables better and more intensive utilisation of information. According to V. Stibic in his book entitled "Personal Documentation for Professionals" a personal documentation system does the following:-

1. prevents information once gained from being lost,
2. intensifies the use of available information resources,
3. improves the organization of knowledge
4. it offers a superb opportunity for creative use of information,
5. for easier linking of facts and ideas, and sometimes even
6. uncovers unseen relations, associations and conclusions.

"All in all : do not regard personal documentation merely as an aid to save time in searching for documents, but primarily as a tool that enables better usage of accumulated

information, that stimulates creative thinking, and that improves your style of intellectual work."

Need for Personal Documentation

Even though several electronic journals are now available - that is journals which are mostly only in machine readable form, there is still no decline in publications in printed forms. A study undertaken in 1973 showed that 7000 papers were published every working day on science and technology alone. The following two tables give an idea of the exponential growth of (1) Number of scientific journals and (2) Number of publications in Astronomy alone

YEAR	NUMBER OF SCIENTIFIC JOURNALS	DECADE	ASTRONOMY: NUMBER OF PUBLICATIONS
1800	100	1600-1610	3
1850	1000	1651-1660	17
1900	10000	1751-1760	311
1950	100000	1851-1860	2721
2000	1000000 (??)	1951-1960	140000

Most persons have a large personal collection of documents of various kinds. Besides this collection a person would also like to have access to useful information which he has read and is stored in various publications not owned by him.

Besides the problem of "Information Explosion" there is another serious problem of "Information Pollution". Due to the "publish-or-perish" syndrome the really important work gets buried amongst the mediocre work. Again it is estimated that only 10% of published scientific and technical literature contains any new facts or ideas.

How can a person keep abreast of the vast quantity of information and developments in a particular field? Regular scanning of secondary periodicals like abstracting and indexing journals also do not solve the problem.

One can obtain information from the library collection, but one would always prefer to refer to ones own personal collection mainly because :- (1) It is immediately available- this is a great boon when information is urgently required (2) One is familiar with the contents of the documents and may have marked important points. This saves a lot of valuable time, since one does not have to spend time in going through several publications.

Mechanised Indexing

Since a professional person would find it difficult to spend time in indexing the documents, and would also not like to learn complicated indexing techniques, we have used a simple mechanised indexing method, whereby index terms from the title are extracted by the computer. A

internal files through which information will be accessed faster. These internal files which essentially form the crux of the system are all direct access files and are named as follows:

1. INDEX.DAT
2. WORD.DAT
3. REPOIN.DAT

When you try to load any document from the data file PERDOC.DAT, all the words from the fields of title, keywords, source, authors and date will be separated. These words will be stored in the file WORD.DAT. Since this is going to be a very big file, to search any word quickly we have created a file INDEX.DAT in which there are 36 records corresponding to 26 alphabetic and 10 numeric characters. Each record has 37 fields corresponding to 36 alphanumeric characters in addition to a blank character. Thus INDEX.DAT looks like a matrix of size 36x37. With the help of first two characters of any word, the file INDEX.DAT will give an integer which is a record number of first word in file WORD.DAT with the same front characters. In WORD.DAT every record has four fields. They consist of word, word type (i.e. code which indicates from which field of the record in PERDOC.DAT this word is selected), a pointer for the next word in the file with the same front characters and an entry pointer in the file REPOIN.DAT. In REPOIN.DAT each field contains an integer which gives the record number in the file PERDOC.DAT (which

actually contains the word to be searched) and a pointer to search the next record containing the same word in the file REPOIN.DAT. The detailed information is given in chapter 4.

III. Retrieval of document (RECRET) :-

This is the most interesting part from the user's point of view, Retrieval from following fields in the record of the document is possible

1. Title, Keywords, Source
2. Authors
3. Date
4. Accession Number

To be more specific retrieval is possible through any of the above fields. One can retrieve a document giving complete information for any field or any word of any field or giving starting substring (minimum two characters) of a word of any field. These words can be in any order and should be separated by comma or blank. The detailed information for this module is given in chapter 5.

IV. Update (UPDATE) :-

While entering the document in the system, often one finds that the user may not have decided the document opinion or the document level precisely. Similarly after thorough reading, the user may like to add some more key-words to the document record. It is also possible that the

user may lose some of the documents already entered in the system. In all such cases the updates of data file PERDOC.DAT as well as the internal files are very important. So we offer updates on the following fields:

1. Delete a record through Accession Number
2. Enter new keywords
3. Change your opinion about the following:
 - a) Document opinion
 - b) Document level

The detail description has been given in chapter 6.

Chapter 3

HOW TO CREATE PERDOC. DAT FILE

In this file, the length of every record is 390 characters. First record is a header record. In order to create a header record, the command 'CREATE HEADER RECORDS' should be given. To be more specific, this command creates the header records for all the files used in the system. It is, therefore, absolutely necessary that one should give this command at the outset. In the header record of the file PERDOC.DAT there are 32 fields. The first field contains the total number of records available in the file PERDOC.DAT, the second the total number of gaps caused due to deletions and the remaining fields contain the actual record number of the deleted records.

The document record in the file consists of eleven fields of different lengths. The information about any document is entered as demanded by the module in a conversational mode. The information should be typed from the first column.

Accession Number:

This field is five-characters long containing integer numbers. Firstly the module will require an accession number of the document. In fact all possible accession numbers for a new document are actually suggested by the system. The user is expected to use one of these numbers. These numbers are generated by the system using the information stored in the header record regarding the gaps in the file PERDOC.DAT

caused by the deletions and the last record number that has been occupied. It is advisable that the user should use the numbers caused by deletions (i.e. smaller numbers) to utilize the disk space properly. It may be noted that the system does not accept any number other than those suggested by the system as accession numbers.

Title:

Title can have maximum 150 characters to be entered in three lines. Maximum 50 characters can be entered in one line. The last character of any title should be '*'. In any line the characters beyond 50 will be ignored.

Keywords:

To give keywords is optional. System will ask the user whether keyword is to be entered or not. If not, enter a character 'N' or just press CR. If you think some additional keywords from which one would like to retrieve this document are needed, then one may enter these words as keywords. Maximum 200 characters for all the keywords put together can be entered in four lines. The characters beyond 50 in any line will be ignored. Different keywords should be separated by commas and the last keyword should be followed by '*'.

Source:

The content of the data describing the source of the document depends on its type. In the case of an article

from a journal, it consists of the journal abbreviation, volume, number and pagination etc. The other non-periodical publication (report, dissertation etc.) may be identified by its place of publication, publisher etc. Maximum 50 characters are used for the description of the source.

It may be mentioned that in the record of the document contained in the file PERDOC.DAT, the fields title, source and keywords are stored together separated by special characters '+'. It is expected that these three fields put together will not exceed 300 characters.

Authors:

For a document one can store maximum four names of authors. One should enter surname followed by initials. Do not enter blank between initials. Maximum 20 characters for any author's name, including the initials can be given.

Physical Form:

The physical form of a document identifies the affiliation of the document to a subcollection. If the volume of this subcollection is very small or if it is arranged according to sequential numbering, then it will be very easy to find that document physically. This field has a length of one character. The codes offered for this field are as follows. B(Book), R(Reprint), L(Letter), T(Computer Output), J(Journal), M(Magnetic Tape) and O(Others).

Date:

This field has a length of six characters. The Date of publication is generally recorded as year-month-day. For example 790915 for 15th September, 1979 should be entered. First two digits for the year is a compulsory field for any document. It may be noted that it should not be greater than the current year. The fields month and day if not available or applicable may be left blank.

Document Opinion:

Following one-letter codes are offered for this field. T(Text), D(Dissertation), R(Reference), N(Notes), O(Others).

If you think that the document can be used as a text or it is a part of a dissertation or a computer output for a project, you may select any one of the relevant codes given above.

Document Level:

While reading a document, often one forms an idea about the level of its content. This may be recorded in the system by selecting any one of the three codes V(Very High), H(High) and L(Low).

After the user answers all the queries regarding the input successfully the system will confirm the accession number and store the document under it. One should put this number on the document immediately. This is very important as it is quite likely that the same information may be entered again under another accession number.

Chapter 4

HOW TO LOAD THE INFORMATION

There are some professionals who receive a limited volume and a small variety of documents and information from external sources. They manage their personal documentation using small, neat and tidy file system. In general they have no problems with documentation and with the storage and retrieval of their documents. Due to the small size of the volume it may even be possible to remember them and find them quickly and reliably without using special techniques.

However, most of the professionals having a large collection of documents feel an intense need for a system that would enable them to find any document in their personal collection easily. For retrieving information the system needs some access points. The access points that a user may like to use are title, keywords, author-names etc. The main aim of this module is to store the information in such a fashion that retrieval from various required access points is equally efficient.

To design such a system we decided to isolate all the words from the fields of title, source, keywords, authors and date. We have then removed all the stop words (see appendix-C), and stored other words in the file WORD.DAT. Since it is going to be a very big file, we have created another

file INDEX.DAT which searches the file WORD.DAT very efficiently. Both these files are direct access files. In order to search a word, its first two characters are used. From the first two characters of the word, we can find the position of the word in WORD.DAT file using the INDEX.DAT file. In WORD.DAT file, against each word there are two pointers. The first pointer points to the next word in WORD.DAT having the same two characters in the front. The second pointer points to the relevant record in the file REPOIN.DAT. The record in the file REPOIN.DAT consists of 10 fields. Each field holds a number which can be used to identify the actual record number in the file PERDOC.DAT and the pointer to the next record in the file REPOIN.DAT.

These three files generated internally at the time of loading form the mainstay of the system. Any document can be retrieved using the permutation and combination of different words from various fields such as title, keywords, authors, date etc. The system has been so designed that even the starting substring (minimum two characters) of the words may be used for retrieval.

The three internal files mentioned above are produced by this module using the file PFRDOC.DAT which is already created by the user. In order to ensure the accuracy of the internal files, it is absolutely necessary that the file PERDOC.DAT is completely error-free. The loading

of the files generally starts for a given accession number. It is also possible to load the documents between a given range of accession numbers. While loading, the list of all the words from which retrieval is possible will be displayed on the screen, and at the end it will give a message that record corresponding to the accession number has been properly loaded. The schematic diagram showing the flow of information leading to loading the document has been given in Fig. 1.1

General Structure of the Internal Files:

In the file INDEX.DAT there are 36 records corresponding to 26 alphabetic letters and 10 numeric characters. In each record there are 37 fields, each having a length of five characters. The length of every record is 185 characters. Size of this file is fixed throughout the use of this system. This file can be conceived as a matrix of 36x37 in which each row corresponds to one record. The rows correspond to alphanumeric characters and the columns correspond to alphanumeric characters and the blank character. For any pair of alphanumeric characters, there is a field containing an integer number which points to the first record of WORD.DAT having a word with the same pair in the front.

In the file WORD.DAT the header record has a length of 160 characters. The other records have a length of 31 characters only. Each of these records have four fields. First field contains a word having maximum length 20

characters. Second field contains one character indicating the code for the field from which the word is selected. The codes are T(Title), keywords, source), A(Authors), D(Date). Third field of length five characters is to store pointer for the next word in the file having the same front character. Last field of length five characters is also a pointer which points to the record in the file REPOIN.DAT.

In the REPCIN.DAT file length of the header record is 160 characters. The header record contains the total number of fields stored, the total number of deleted fields and the actual numbers of the deleted fields. All other records have ten fields each of length twelve characters. Every field consists of an integer number which is a combination of accession number and the pointer to the next field containing the same word in the file REPOIN.DAT. To be explicit the accession number is stored in the rightmost 17 bits and the pointer is stored in the rest of the bits of the single word of 36 bits.

We now illustrate the process of loading by taking a simple example. Consider the following record in the file PERDOC.DAT.

ACCNO	:	50
TITLE	:	Programming in PROLOG
AUTHOR	:	Clocks in WF Mettish CS
SOURCE	:	Springer-Verlag
DATE	:	850000
KEYWORDS	:	Computers, Artificial Intelligen

First the system will isolate the words and remove the stop words like 'in' from the fields of title, keywords and source. The words to be stored in three internal files are thus PROGRAMMING, PROLOG, CLOCKSIN WF, METTISH CS, SPRINGER-VERLAG, 850000, COMPUTERS, ARTIFICIAL AND INTELLIGENCE. First of all the system will read the 18th (corresponding to letter R) field of 16th (corresponding to P) record of the file INDEX.DAT. If the content is zero, it implies that there is no word in the system starting with letters PR. In this case the system will try to place it in a position by using the contents of the header record. If the header record says that there are 200 words and no gap in between them, then the new word will be obviously placed in the record number 201. The record number is subsequently stored in the INDEX.DAT file in the appropriate field. Before the word is actually stored in the WORD.DAT file, it also needs to find the entry point in the file REPOIN.DAT. If we assume that there is a gap in REPOIN.DAT file at the field number 300, it will be filled by an integer number formed using the accession number 50 and the next pointer which is zero at this stage. Now the system will store the word in WORD.DAT in the record number 201 along with entry point 300 in the fourth field, a character T in the second field and again a number 201 in the third field.

Now let us take the word PROLOG. Since 18th field of 16th record has integer 201, system will read 201th record of the file WORD.DAT and compare with the word

PROGRAMMING which is stored in it. Since it does not match with it the system will read third field of that record. Since it is also 201, it implies that there is no other word starting with PR. The system will now find a place for the word 'PROLOG' in the record number 202 assuming that there is no gap in the file. It may be noted that there is no separate entry for this word in the file INDEX.DAT. The contents of the other fields in the WORD.DAT and REPOIN.DAT are determined as described above.

Now let us take a word COMPUTERS. If the 15th field of the 3rd record has an integer value 100, system will read 100th record of file WORD.DAT. If the word stored in that record is COMPUTERS then it will read fourth field of that record, say that number is 155. Then the system will read 5th field of the 17th record in the file REPOIN.DAT. It may be noted here that each record in the file stores 10 fields. The contents of the field are separated into two components viz accession number and the next pointer. If the pointer for the next record is zero and the accession number of the document is 25, the system will read the header record of REPOIN.DAT. If there are no gaps and total number of fields are 310, system will combine two integers 311 and record number 25 in a single word and update the 5th field of the 17th record in REPOIN.DAT. System will combine numbers 0 and 50 in a single word and store that integer in the first field of 33rd record of REPOIN.DAT.

Similarly all other words will be stored in the three internal files depending on the contents of the files.

Chapter 5

HOW TO RETRIEVE INFORMATION

As mentioned in previous chapters retrieval is possible using the following four modes. Each of these modes contains the following fields used for retrieval.

1. Title, Keywords, Source
2. Authors
3. Date
4. Accession Number

The retrieval from the first three modes is possible only if that document is already loaded in the system. The retrieval using the accession number is possible using the datafile PERDOC.DAT alone. When an attempt is made to retrieve information using this module, one will see the menu of the four modes as given above. One can select either the sequence numbers corresponding to first three modes separated by comma in any combination or the mode number 4 exclusively for ones query. If more than one mode is selected, then they will be connected by AND operator by the system, implicitly. The schematic diagram showing the flow of information leading to retrieval of the document is given in figure 1. We will now describe each mode in detail.

Mode 1:

This is a very versatile mode which uses any word or the starting substrings of the words appearing in the

fields title, keywords and source. Furthermore these words or starting substrings can appear in any order. All these words are connected by an AND operator implicitly. Consequently for the successful utilization of this mode, all the words and the starting substrings should be typed correctly. These words belonging to three different fields are accepted in maximum six lines, each containing maximum 50 characters. The words in the lines are separated by blanks.

Mode 2 :

In this mode the name of the author is used for retrieval. The name consists of surname and a group of initials separated by blank. One can even use the correct starting substring of the surname without initials. When there is more than one author, the names are separated by commas. In this case system will ask for the connecting operator to be used for different author-names. The connecting operator can be either AND or OR. Both operators cannot be used simultaneously.

Mode 3 :

This mode uses the Date information for the retrieval. The information in general consists of year (2 digit integer), month (2 digit integer) and day (2 digit integer) in the same order. While the year information is compulsory, the other fields may be left blank if not available or known.

Mode 4 :

The accession number being the primary key of the document record in the file PERDOC.DAT, it enables one to retrieve the document uniquely. Consequently when one uses accession numbers for retrieval, then one need not use other modes simultaneously. In this mode one can either use a single accession number (5 digit integer) or a range of accession numbers (i.e. 112 to 148) to retrieve the required documents.

After displaying the information retrieved in response to a query, the system will ask if a hardcopy for the same is needed. If required, it will be stored in the file PRINT.OUT which can be printed subsequently at the end of the session.

Chapter 6

HOW TO UPDATE INFORMATION

Since this is a personal documentation system, it is expected that ~~only a few updates are necessary~~. The system is so designed that, while entering data, it will echo the inputted data and ask for approval. So even at that stage some of the typing errors can be corrected.

If information for many documents is being entered, one may not be able to find or correct all the errors that may creep in while entering the data. In this case it may be advisable that after entering all the data, the data is retrieved through Accession Numbers and a hard copy of all the documents is taken for manual checking. If any error is found one has to delete the corresponding record and reenter the same properly. This way it can be ensured that the document records in the file PERDOC.DAT are absolutely correct.

However it is very much possible that, while entering a document one may not have decided some of the fields such as document opinion or document level precisely. In other words this field may need to be changed subsequently. After proper reading of the document one may like to add some additional keywords from which the document may be retrieved. One may also like to take into account the possibility that the document already entered in the system has been lost or missing. In all such cases the internal files of the system need to be updated.

-24-

This module will update the system database files as per the menu given below for any given accession number.

1. Delete a record through accession numbers.
2. Enter new keywords.
3. Change your opinion about the document and its level.

The user can select the number against the required update in the menu.

If details of a document, using an accession number is deleted then the system also deletes its various entries in the internal files.

For entering new keywords, the system will first inform the user the balance field space available. The user may utilize this space by entering the new keywords to be separated by comma.

The third mode of update will enable the user to change the document opinion or the document level or both in a conversational mode.

Since all the data as well as internal files are direct access files, no physical deletion will take place. Instead a delete flag will be placed in the record for deleting the record logically from all the files. After deleting the record, the system will simultaneously update their header records properly to keep track of gaps caused by them so that they can be utilized for entering new records.

CONCLUSION

As stated earlier the Personal Documentation System which has been developed is very user friendly. The system automatically prompts the input of all types of documents hence inputting data is also simple and quick. In order to save cost and time required for manual indexing, very simple mechanised indexing has been introduced. Provision of key-words or descriptors, enable the user to cover the subject content of the documents. Keywords also take care of pseudo-titles, which do not convey the real coverage of a document. For example a title "Betrayers of Truth" really deals with fraud in science.

Provision of a list of stop words also helps the user. If a persons subject of interest is "Personal Computers", then he could add this term in his list of stop words, for most of his collection will deal with this subject.

In the absence of a mechanised system, most persons use a simple card index system, which is their main retrieval tool. This has severe limitations because at the most one card is prepared, which could be arranged in an authorwise sequence. This will not permit subject retrieval. Even if trouble is taken to prepare several cards (one for each subject) and the cards are arranged subjectwise; then too, in a manual system it is not possible to retrieve documents which contain a combination of 2 to 3 keywords.

This system can also be used for "Group Documentation" i.e. a document retrieval system, comprising of the total personal collections of all the members of one group, having the same subject interest. The personal files of each member need not be merged into one centralised file, as it would be easier to have access from individual collections.

A mechanised personal documentation enables better organisation of ones collection of documents. Access to information is not only quick and easy, but it permits effective utilization of accumulated knowledge. It also provides a large number of different ways by which information can be accessed.

Reference:

Stibic, V.

Personal Documentation for Professionals
1980, Amsterdam : North Holland Pub. Co.

APPENDIX-A

Structure of the System

There are 28 different subroutines in the system MYLIB.

List of all the subroutines in MYLIB is given below.

- | | |
|------------|------------|
| 1. PERDOC | 15. PART |
| 2. LOADDB | 16. LDJPL |
| 3. RECRET | 17. DQSORT |
| 4. UPDATE | 18. SEARCH |
| 5. INSQR | 19. TRUN |
| 6. UPPER | 20. IQSORT |
| 7. YORN | 21. DISPLY |
| 8. ACLEN | 22. PRIDIS |
| 9. DATVAL | 23. DELACC |
| 10. SEPRAT | 24. DELSER |
| 11. ALPSER | 25. ADDKY |
| 12. WCHAIN | 26. DUPL |
| 13. RCHAIN | 27. HEDREC |
| 14. JOIN | 28. MYHELP |

Out of these 28 subroutines four subroutines PERDOC, LOADDB, RECRET, and UPDATE represent four major modules of the system. Each one of them calls different subroutines. The list of all subroutines in each module and the main program MYLIB is given below :

MYLIB:

1. MYHELP
2. HEDREC
3. PEIDCC
4. LOADDB
5. RECRET
6. UPDATE

PERDOC :

1. DISPLY
2. INSQUR
 - 2.1 YORN
 - 2.2 ACLEN
 - 2.3 UPPER
 - 2.4 DATVAL
3. YORN

LOADDB :

1. UPPER
2. SEPRAT
3. LDUPL
 - 3.1 DQSORT
4. ALPSEK
 - 4.1 WCHAIN
 - 4.1.1 JOIN
 - 4.1.2 RCHAIN

4.1.2.1 PART

4.1.2.2 JOIN

4.2 JOIN

5. YORN

RECRET :

1. PRIDIS

1.1 UPPER

1.2 DISPLY

1.3 YORN

2. UPPER

3. SEPRAT

4. LDUPL

4.1 DQSORT

5. SEARCH

5.1 PART

6. TRUN

6.1 IQSORT

7. DUPL

8. DATVAL

9. YORN

10. DISPLY

SECRET

UPDATE :

- 1. DISPLY
- 2. UPPER
- 3. YORN
- 4. ADDKY

- 4.1 SEPRAT
- 4.2 ACLEN
- 4.3 YORN
- 4.4 LDUPL
- 4.5 UPPER
- 4.6 ALPSER

4.6.1 JOIN

4.6.2 WCHAIN

4.6.2.1 JOIN

4.6.2.2 RCHAIN

4.6.2.2.1 PART

4.6.2.2.2 JOIN

5. DELACC

- 5.1 SEPRAT
- 5.2 LDUPL
- 5.3 UPPER
- 5.4 DELSER

5.4.1 PART

5.4.2 JOIN

APPENDIX B

Instructions to use MYLIB

1. Operate the system using the command
RUN MYLIB (231, 104)
2. Type the message HELP to get additional information
and instructions regarding the usage of the system.
3. If you are entering the first document of the system,
type the following command without fail.
CREATE HEADER RECORDS
4. Subsequently the following menu will appear on the
screen.
 1. Enter a new document
 2. Loading into database
 3. Retrieval of any document
 4. Update of any information
5. Select item number 1 in the menu to enter the documents.
These documents will be stored in the file
PERDOC.DAT.
6. Select item number 2 in the menu to load the records
in the file PERDOC.DAT in various internal files for
efficient retrieval using various fields.

7. Now the system is ready for retrieval and updates. Select item number 3 for retrieval and 4 for update. For retrieval one may use any word or starting substring of the words contained in the files of title, authors, keywords, source and date in the order as requested by the system.

APPENDIX-C

List of stop words

MYLIE removes following stop words from the data as they are not important from the retrieval point of view.

IS	ARE	A
AN	THE	AT
ALL	TO	AND
IT	BY	OF
IN	THAT	WAS
HE	FOR	WITH
HIS	ON	BE
I	THIS	HAD
NOT	BUT	WHICH
OR	FROM	SUCH
MAY	AS	a blank

TYPICAL RETRIEVAL SESSION

Retrieval on following fields is offered:

1. Title & keywords & source
2. Authors
3. Date of publication
4. Accession number

Enter Numbers of your choice separated by comma

• 1

Enter, substring of the title, keywords & source followed by* such that all words are separated by blanks. You are given six lines, enter maximum 50 characters in one line

1234550

• Indexing Information storage and retrieval

This information will be considered. If any change enter N (in first column) else press (CR). Search will be performed on following words:

Indexing, Information, Retrieval, Storage

- There are 9 documents containing all 4 descriptors. Do you want them on the screen? Enter N for no else press (CR)

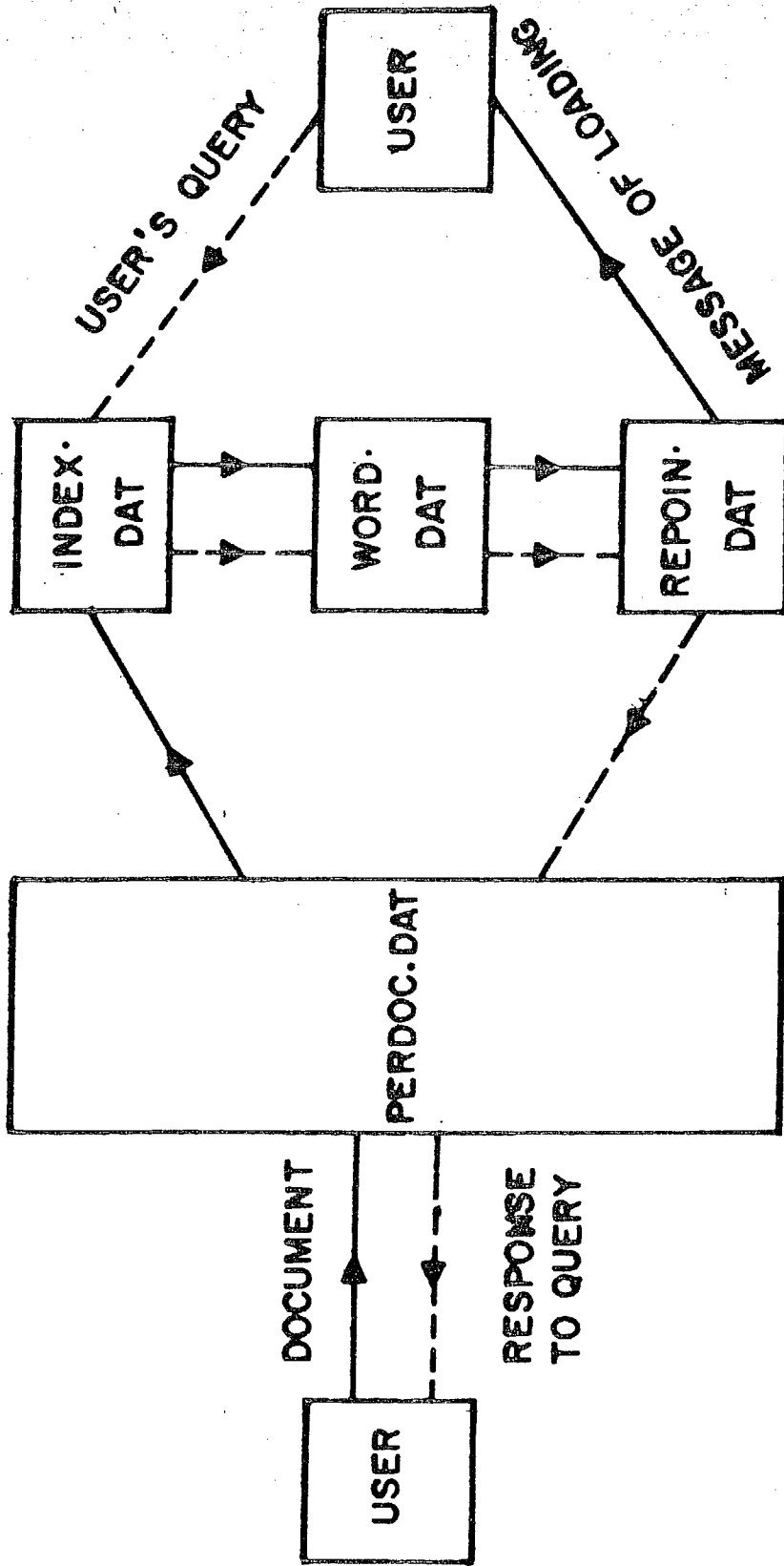
• Press (CR)

ACC NO: TITLE:
----10 VOCABULARY CONTROL IN INFORMATION RETRIEVAL SYSTEMS
AUTHOR:
LANCASTER FW
KEYWORDS:
INFORMATION STORAGE & RETRIEVAL, INDEXING, THESAURI

Document Opinion: DATE:720000 SOURCE : ADVANCES IN LIBRARIANSHIP V
Document Level: Physical Form : REPRINT

ACC NO: TITLE:
----28 A PERSONAL COMPUTERISED LITERATURE RETRIEVAL SYSTEM
AUTHOR:
CONNOLLY J REILLY V HEGARTY T
KEYWORDS:
PERSONAL DOCUMENTATION SYSTEM, INFORMATION STORAGE AND RETRIEVAL, AUTOMATIC INDEXING

Document Opinion: DATE:820000 SOURCE : JNL OF INFORMATION SCIENCE V4 1982 PP 97
Document Level: Physical Form : REPRINT



--- RETRIEVING
 ——— LOADING

DATAFLOW IN MYLIB SYSTEM