Atlanta University Center
DigitalCommons@Robert W. Woodruff Library, Atlanta University Center

ETD Collection for AUC Robert W. Woodruff Library

7-1-1982

The software package for the Air Force emergency crew escape data

Dhawan Pongdee
Atlanta University

Follow this and additional works at: http://digitalcommons.auctr.edu/dissertations

 Part of the Applied Mathematics Commons, and the Mathematics Commons

Recommended Citation
Pongdee, Dhawan, "The software package for the Air Force emergency crew escape data" (1982). *ETD Collection for AUC Robert W. Woodruff Library.* Paper 2696.

This Thesis is brought to you for free and open access by DigitalCommons@Robert W. Woodruff Library, Atlanta University Center. It has been accepted for inclusion in ETD Collection for AUC Robert W. Woodruff Library by an authorized administrator of DigitalCommons@Robert W. Woodruff Library, Atlanta University Center. For more information, please contact cwiseman@auctr.edu.

THE SOFTWARE PACKAGE FOR THE AIR FORCE EMERGENCY CREW ESCAPE DATA

A THESIS

SUBMITTED TO THE FACULTY OF ATLANTA UNIVERSITY

FOR FULFILLMENT OF THE REQUIREMENT FOR

THE DEGREE OF MASTER OF SCIENCE

BY

DHAWAN PONGDEE

DEPARTMENT OF MATHEMATICAL SCIENCES

ATLANTA , GEORGIA

JULY , 1982

R=VI P=49

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I wish to express my deepest appreciation, especially for my advisor, Dr. Arthur M.Jones. This thesis could not be accomplished without his support and patience. Furthermore, I would like to extend my respect to the following Mathematical Department instructors:

Dr.Benjamin J. Martin

Dr. Johnny L. Houston

Dr. Nazir A. Warsi

Dr. Ronald Biggers

Dr. Bennett Setzer

Prof. Grover Simmons

Those instructors have been extremely helpful to me for all the time that I have been Atlanta University student.

ABSTRACT

COMPUTER SCIENCE

Dhawan Pongdee B.E. King Mongkut's Institute of Technology, Bangkok, Thailand, 1977.

THE SOFTWARE PACKAGE FOR THE AIR FORCE EMERGENCY CREW ESCAPE DATA.

Advisor : Dr. Arthur M. Jones

Thesis date July 1982

Because record-keeping and decision-making in industry and government are increasingly based on information stored in computer-processable data bases, the need for improved hardware and software technology for building, managing and using these data bases is becoming ever more important.

In order to gain a full understanding of the tools and problems of this important computer application, the author made this attempt to develop and document his own data base software product.

INTRODUCTION

Since 1976 US Air Force experience in noncombat emergency aircraft ejections is indicative of an upward trend in the number of ejections accompanied by a decline in the annual survival rate. It is general knowledge among cognizant Air Force personnel that this certainly has not been the result of inattention or technological stagnation. The rate of survival seems intractable despite dramatic advances in the state-of-the-art of escape system. Life Science Group Directorate of Aeorspace Safety, Norton AFB, California has established an enviable record of compiling and disseminating data on current trends among cognizant agencies. This thesis was conducted in support of Dr. Arthur M. Jones' effort to strengthen and refine their effort by providing comprehensive analysis of those data; emphasis will be put on studying joint effects of causal factors on survival rates in contrast to the traditional approach of viewing such effects on a one-at-a-time basis. Specificially, this project supported three of Dr. Jones' goals:

1. to conduct a literature search for new or updated information on the storage and maintenance of a computerized data bank.

2. to develop a computerized data bank consisting US Air Force emergency ejection data.

3. to formalize a data management system for that data bank.

The application program, about which this thesis is written, has the ability to manage a data bank in such ways as retrieving, updating, inserting and deleting. One of the more important functional aspects of this program is the data retrieval. Flexibility was a principal objective in the design of data retrieval. Data may be retrieved from the data base in three ways:

1. retrieval of the whole file.
2. retrieval of the whole file with atmost 4 selected fields of each record.
3. retrieval of all records within the range of fields qualifiers ($\geqslant$ lower bound, $\leqslant$ upper bound).

The output of the above three ways provides two functions:

1. list to terminal any existing user file or the master file.
2. create a copy or a subset of the master file.

It should be noted here that the design of this application program was constrained by Dr. Jones'requirements.

Certain compromises in theoretical and technical consideration were made in the interest of simplicity,modularity and maintainability. The author would have preferred to have designed a more sophisticated software product, in term of efficiency and economy but deferred to Dr. Jones.

This thesis consists of three chapters. Chapter I contains some discussion about data base concepts. Chapter II provides a program overview and details of subroutines,including flowcharts. Chapter III contains recommendation for future upgrades. Also, there are appendices A and B. Appendix A contains glossary and symbol representation, and appendix B contains figures about computer system,etc. References are attached at the end.

CHAPTER I

-- Data Base Concepts

-- Storage Media of Data bases

-- Data Base Management System (DBMS)

-- An Application Programs

CHAPTER I

DATA BASE CONCEPTS

Data Base Concepts

In this chapter the author will give a general overview of the topic of data base. The purpose of this is to acquaint the reader with my view of a data base. How this project fitted to the concepts will be discussed in the chapter II.

The words to describe data base have varied from one authority to another. Generally data base may be defined as a collection of interrelated data stored together on disk, drums, or other secondary storage media. The data base may be designed for batch processing, on-line processing, real-time processing, in-line processing, or the above processing combination. The data are stored so that they are independent of programs which use the data.

The data base contains the data for many users, not just for one, which in turn implies that any one user may be concerned with just a small portion of it; moreover, different users'portions may overlap in various ways that is, individual pieces of data may be shared by many different users.

"The cost of data-storage hardware is dropping more rapidly than other costs in data processing. It will soon become cheaper to store data on computer files than to store them on paper."[1]

To make use of the huge quantities of data that are being stored, two system facilities are needed in addition to the storage. These are data transmission- the ability to access the data base from remote locations where its information is needed, by means of telecommunication and man-computer dialogues, which enable the users to make inquires, browse in the files, modify the data stored, add new data, or solve problems which use the data.

A data base exists as a physical collection of data organized to reflect the needs of the user. The difficulty in building effective data base system is not in the mechanics of construction, but in the intelligent design and purposeful use of the data base.

[1] James Martin, Computer Data-Base Organization, 2nd. edition, Prentice-Hall, Inc., Englewood Cliffs, N.J., p. 2.

There are two important components in the development of a data base system that must be successfully implemented

1. Data base design
2. Program design

Since data bases have a very close relationship to the program that support them, decisions concerning the structuring of data must be filtered through programming considerations.

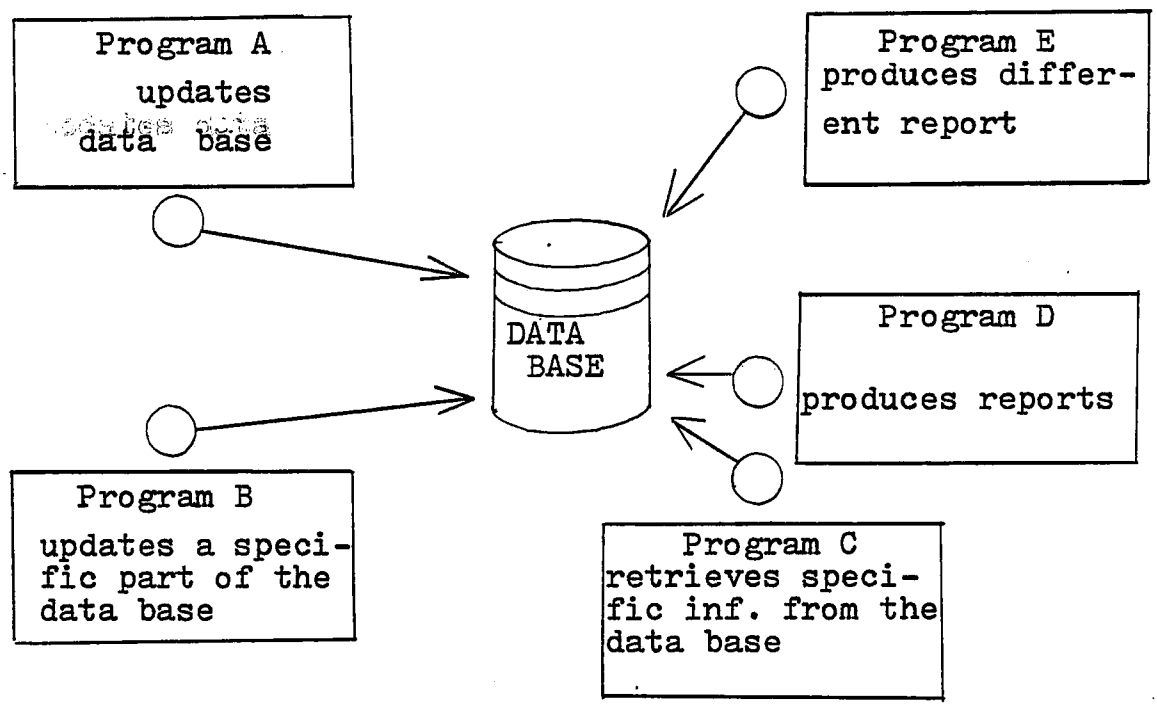


Fig.1 The activity of the data base is performed by several programs each of which has a separate function.

The objective of the data base system is to make application development easier, cheaper, faster and more flexible.

Storage Media of Data Bases

Data bases usually store in direct-access devices (disk or drum), but they may be stored in sequential storage (tape) as well. A major disadvantage of sequential storage (tape) is that there is only one way to search the data which is sequential method (indirect access) and take much time. However magnetic tape is likely to remain the principal storage medium for many years, because tape is low cost and high retention qualities.

Data Base Management System (DBMS)

" A software that handles the storage and retrieval of the records in the data base. The DBMS cannot exist without a data base. The DBMS is the active partner and the data base is the passive one."[2]

A data base management system is necessary to implement the data base concepts. With DBMS, a user can access data and derive the benefits of the data base concept.

[2] Anthony Ralston, Encyclopedia of Computer Science, 1st. ed., Van Nostrand Reinhold Company. p.391.

An Application Programs

Application programs which utilize a data base are able to perform such functions as : retrieving, updating, inserting and deleting.

Many characteristics that programmers have incorperated in programs used to manage data bases are

1. minimization of processing time;
2. effective use of memory space;
3. maximization of flexibility;
4. ease of application development.

CHAPTER II

DATA SYSTEM MANAGEMENT PROGRAM

--Program Environments

--Program Characteristics

--Data

--Overall System

--Program Descriptions

CHAPTER II

DATA SYSTEM MANAGEMENT PROGRAM

Program Environments

The application program was written to operate in the following hardware/software environment.

I. Hardware

The program is designed to run on the VAX 11/780 system which is built by DEC (Digital Equipment Corporation), under the VMS operating system. This system has 1.5 M bytes of main memory, 256 M bytes on disk (RM 05) and a tape drive (TU 77).

II. Software

The PASCAL language is used to manage the data base. There are several interactive versions of PASCAL, but the one that is used for this project is the PASCAL which was developed by DEC to use on the VAX system.

Program Characteristics

The main objective of this program is flexibility. Flexibility is especially important because future changes in record format may occur without forwarning. Flexibility will be very useful when changes make program modification necessary.

In the event that one wishes to extend the number of columns of a given field in each record, one must add the required number of new column to the value designated in " CONST " part. For example, if the user wishes to amend the field width of " Attitude " from 5 to 7, then one need only change the value of ALCOL from 5 to 7.

Whenever extra fields are needed in the future, new variables must be declared in the " CONST " part and "TYPE" part. Also, those new variables must be added to the every " READLN " and " WRITELN " statements about which read or write the value of fields.

Data

The data for which this program was written, was compiled from several logical files based on such fields of interest as injury, ejection, etc. and merge into one physical file.

The logical lay-out of record as below:

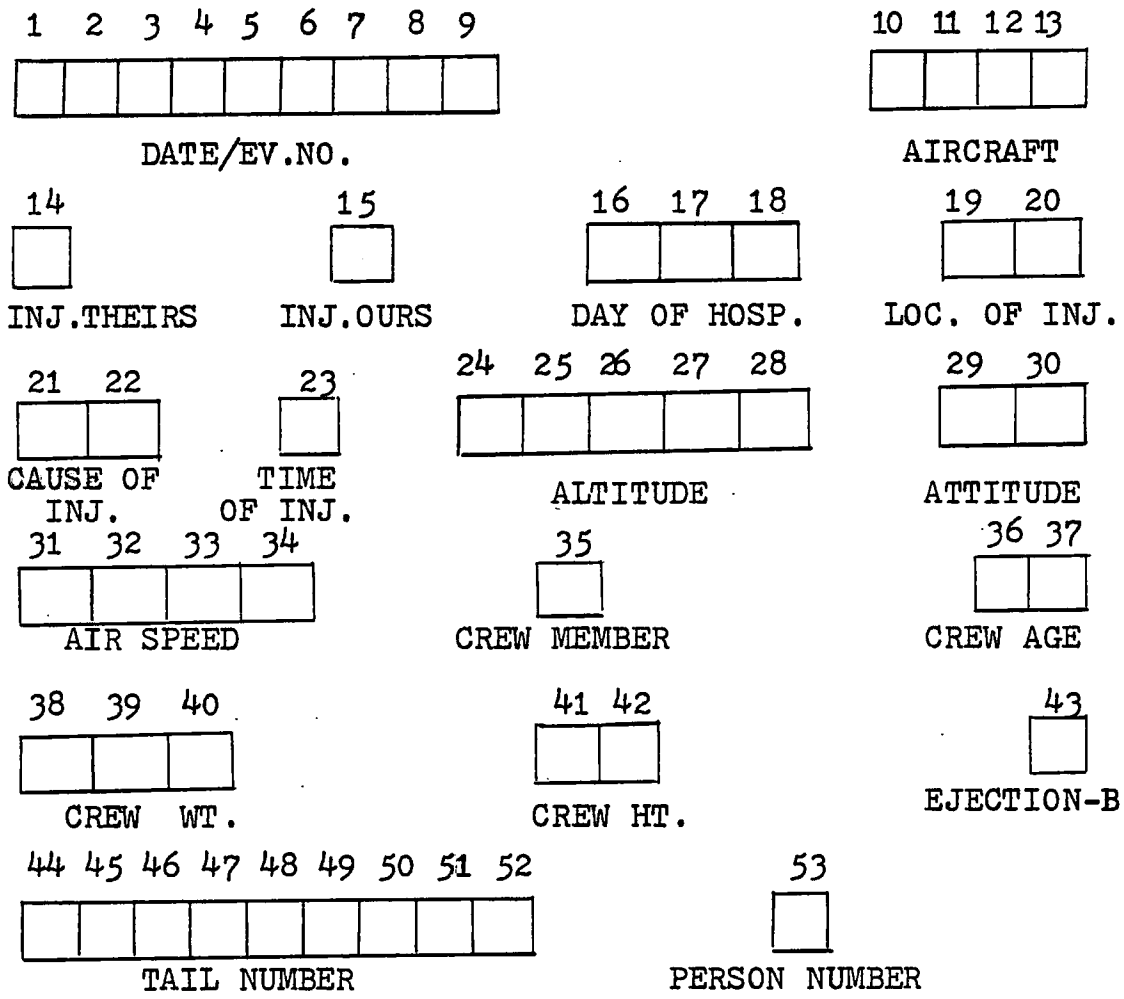


Fig. 2 Record Lay-out

Record Map

Field no.	Columns no.	Field name
1	01-09	DATE/EV.NO.
2	10-13	AIRCRAFT
3	14-14	INJ.THEIRS
4	15-15	INJ.OURS
5	16-18	DAY OF HOSP.
6	19-20	LOC. OF INJ.
7	21-22	CAUSE OF INJ.
8	23-23	TIME OF INJ.
9	24-28	ALTITUDE
10	29-30	ATTITUDE
11	31-34	AIR SPEED
12	35-35	CREW MEMBER
13	36-37	CREW AGE
14	38-40	CREW WEIGHT
15	41-42	CREW HEIGHT
16	43-43	EJECTION-B
17	44-52	TAIL NUMBER
18	53-53	PERSON NUMBER

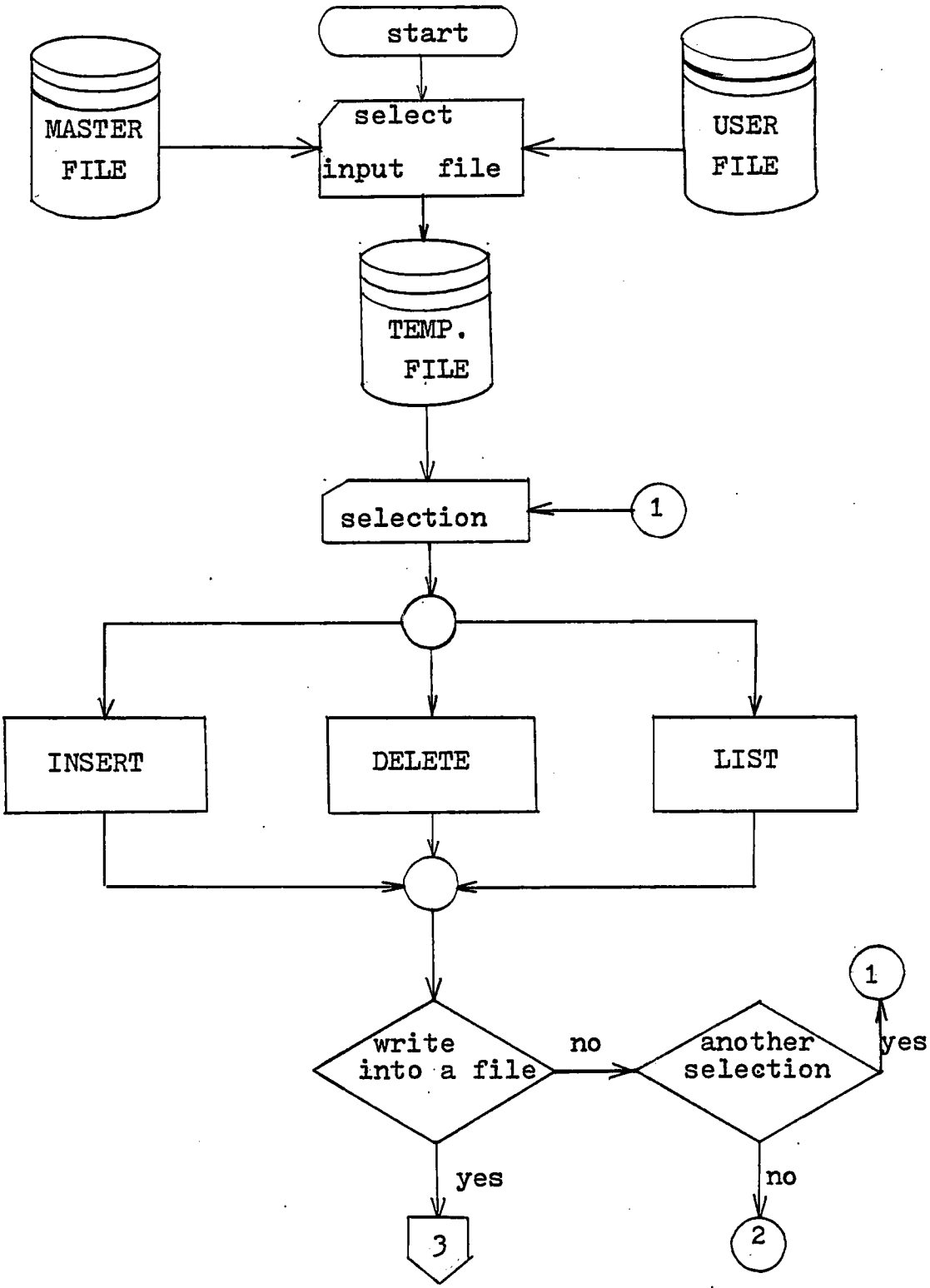


Fig. 3 OVERALL SYSTEM

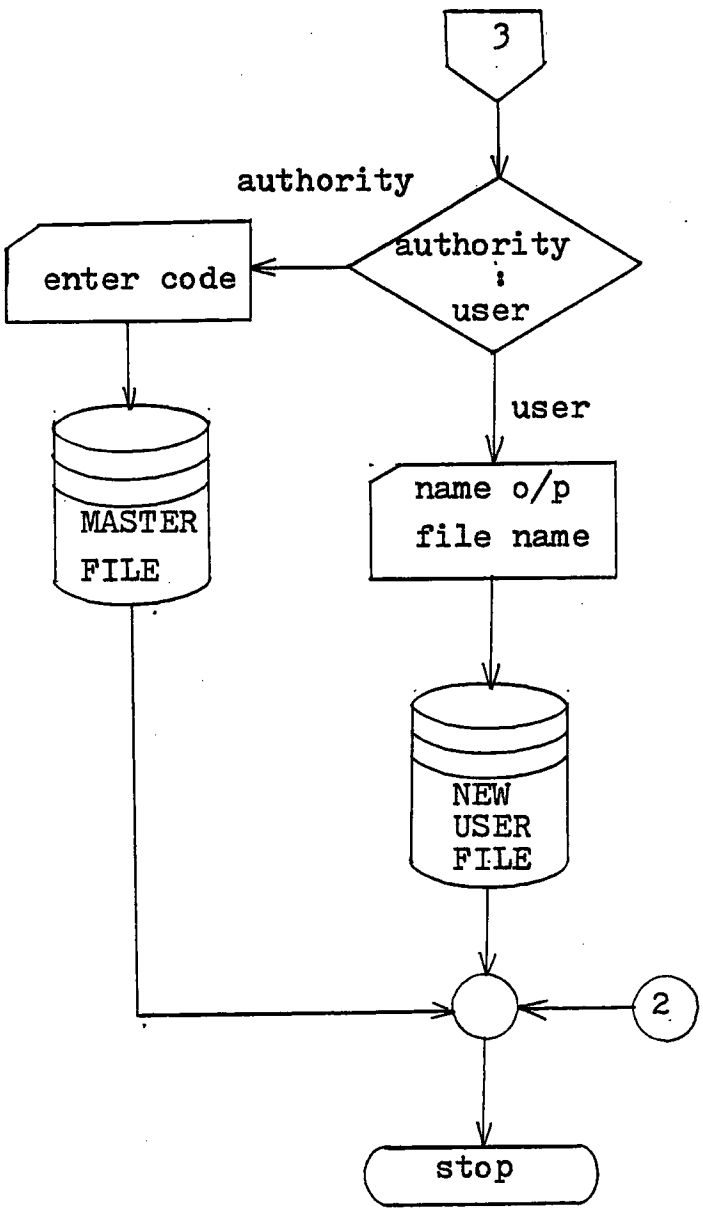


Fig. 3 OVERALL SYSTEM (cont.)

Program Descriptions

1. Program Components

Since there are many actions to deal with the data file, such as insertion of new data, deletion of unwanted data or listing desired information, the dynamic data structures was implemented to write this computer program.

The dynamic data structures is a data structure that changes in size during the execution of the program. Static structure, on the other hand, remains fixed in size through out its lifetime.

Some properties of dynamic data structures are:

-- the declaration of the value of listsize is not required
-- the addition of components at anyplace of the list does not require a re-sorting of the components
-- holes are not left after deleting a component.

The simplest type of dynamic data structure is the linked list which was used to write this thesis program. It provides a solution to the problem of maintaining a list to which components may be added or deleted at random.

Using the linked list only the first component of the list is readily accessible; therefore sequential searching was used for the searching method.

2. Procedure INSERT

Procedure INSERT is designed for adding a new record to the tempory file. With the linked list property, a new record will be inserted at anyplace of the list without re-sorting. All records were sorted by DATE/EV.NO. After insertion, there are options whether or not a new file will be updated. The authority may update the master file by using a password code and the user only update his own file.

The flowchart for insertion is shown in fig. 3.

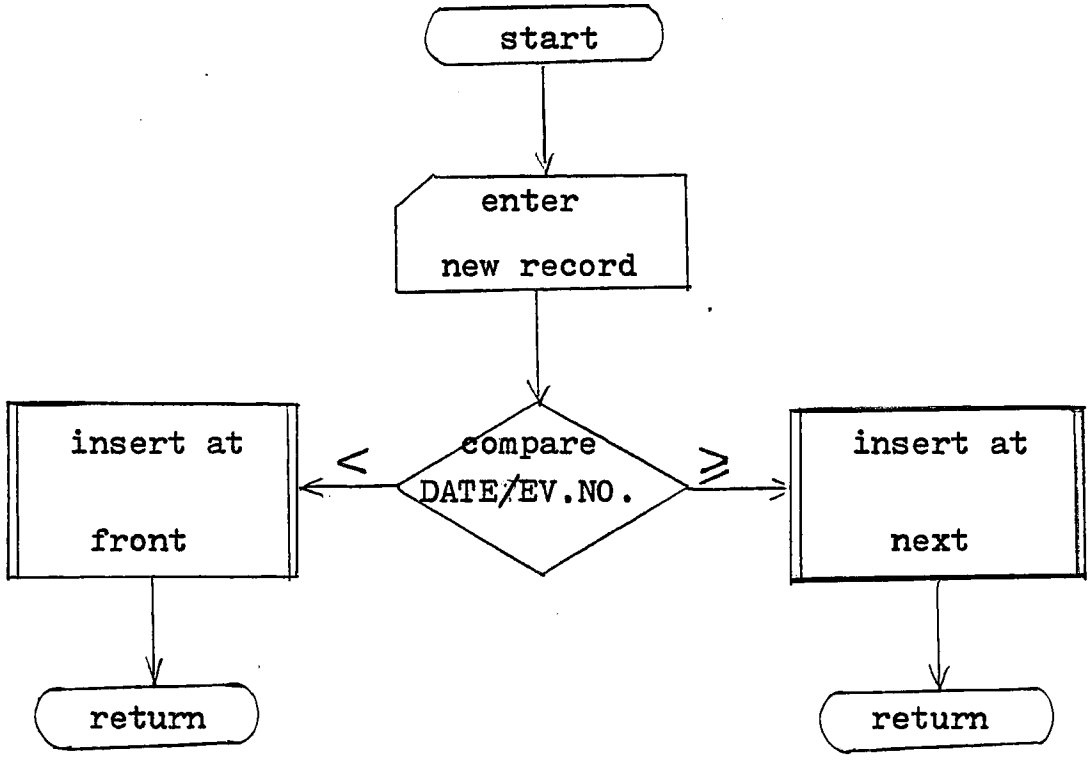


Fig. 4 Insert's flowchart

3. Procedure DELETE

Procedure DELETE utilizes DATE/EV.NO. and PERSON NO. as a key which is unique key. After input DATE/EV.NO. and PERSON NO, the system will search a record by sequential. If the record has the value of DATE/EV.NO. and PERSON NO. corespond with the key, that record will be deleted out of the temporary file. There will not be hole after deleting, because linked list was used in procedure DELETE. There are the same options as procedure INSERT about updating the file. Its flowchart is shown in fig. 4.

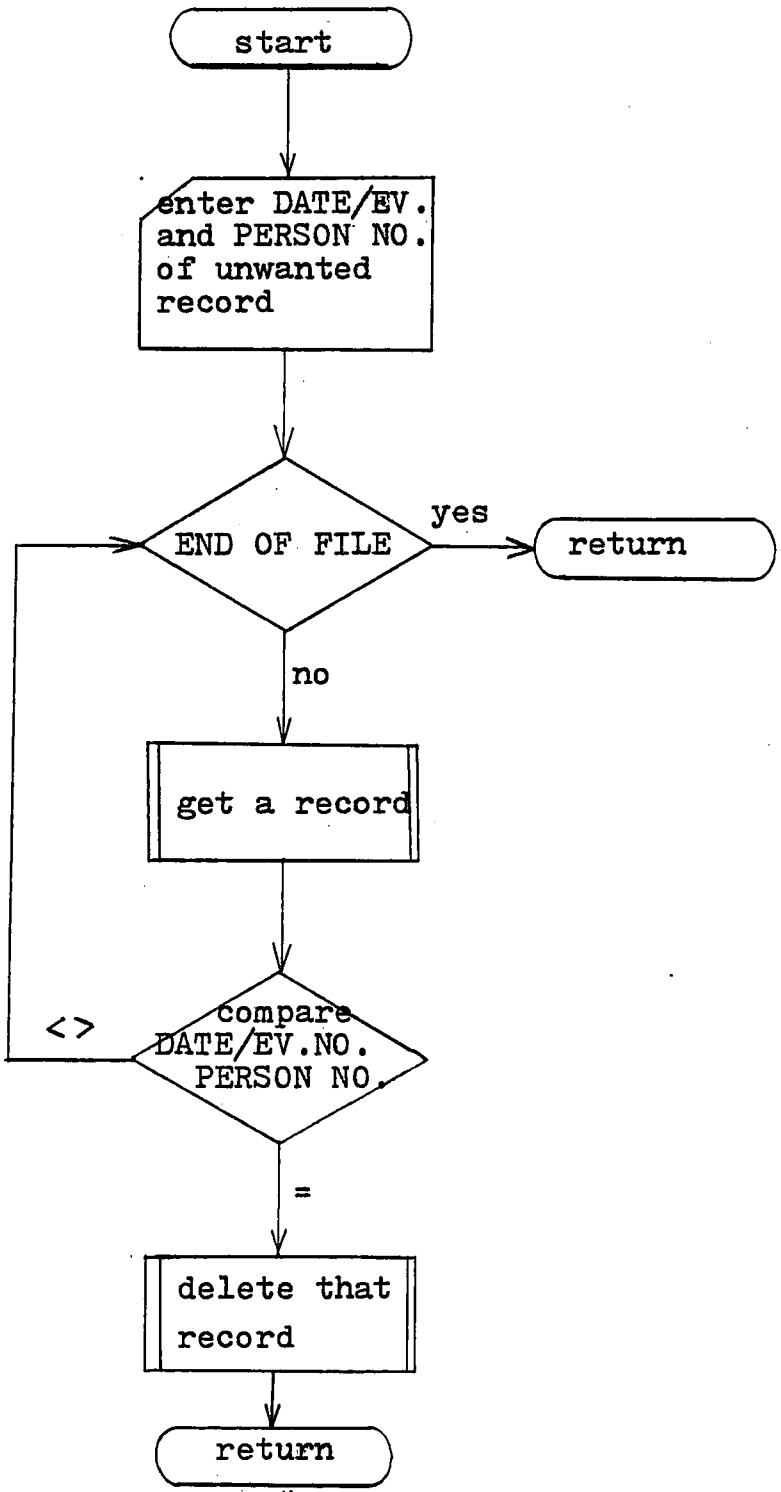


Fig. 5 Flowchart of Deleting.

4. Procedure LIST

Program LIST is designed to retrieve data in three different ways:

1. list the whole file
2. list the whole file with atmost 4 selected fields of each record
3. list all records within the range of fields qualifiers ($\geqslant$ lower bound, $\leqslant$ upper bound).

The output of the above three ways may be written into a new file or on the terminal.

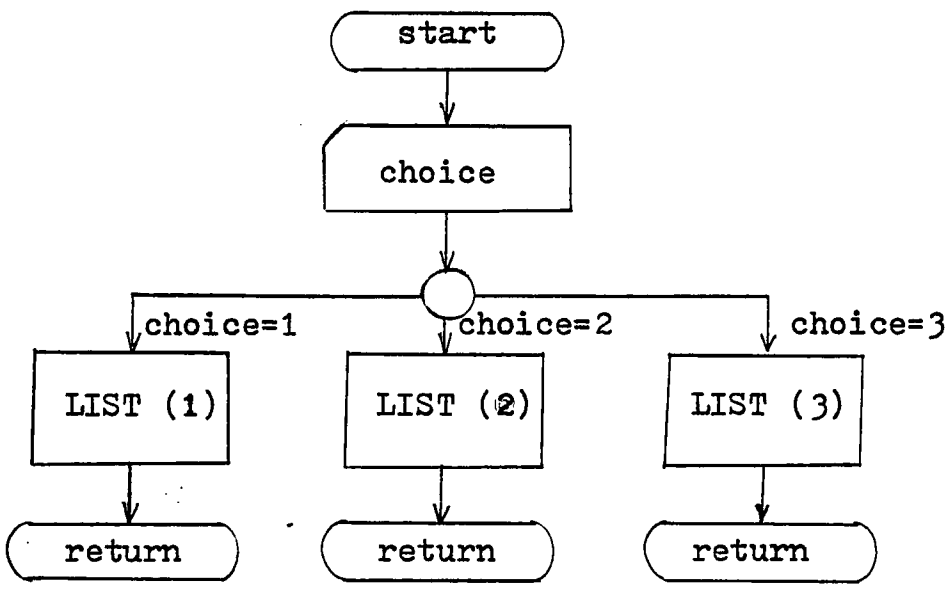


Fig. 6 LIST's flowchart

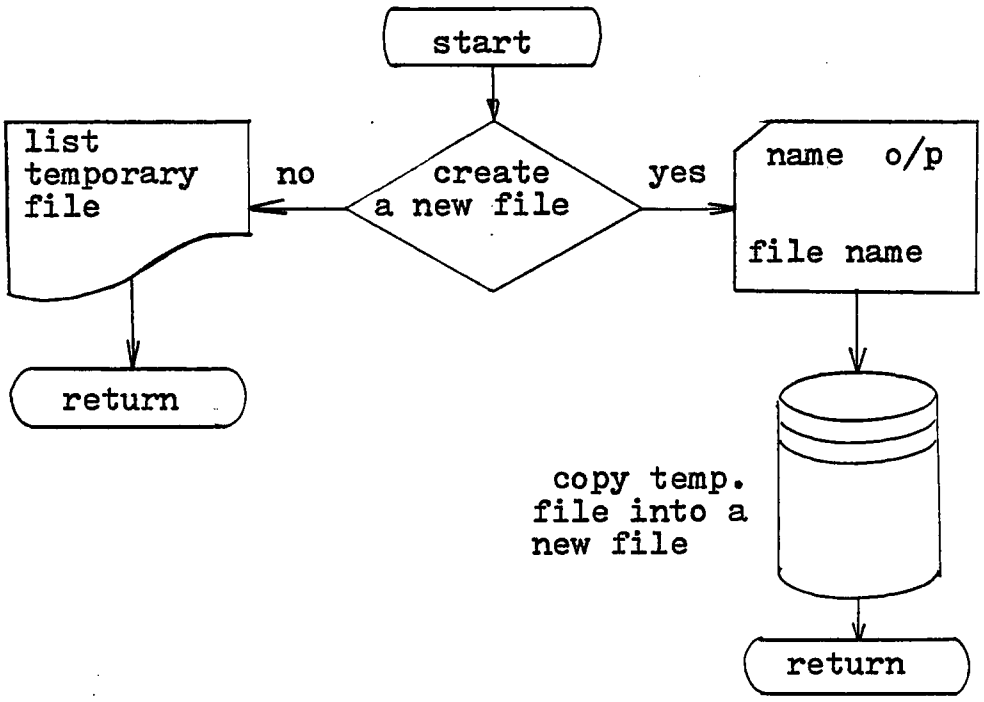


Fig. 7 Choice of List = 1

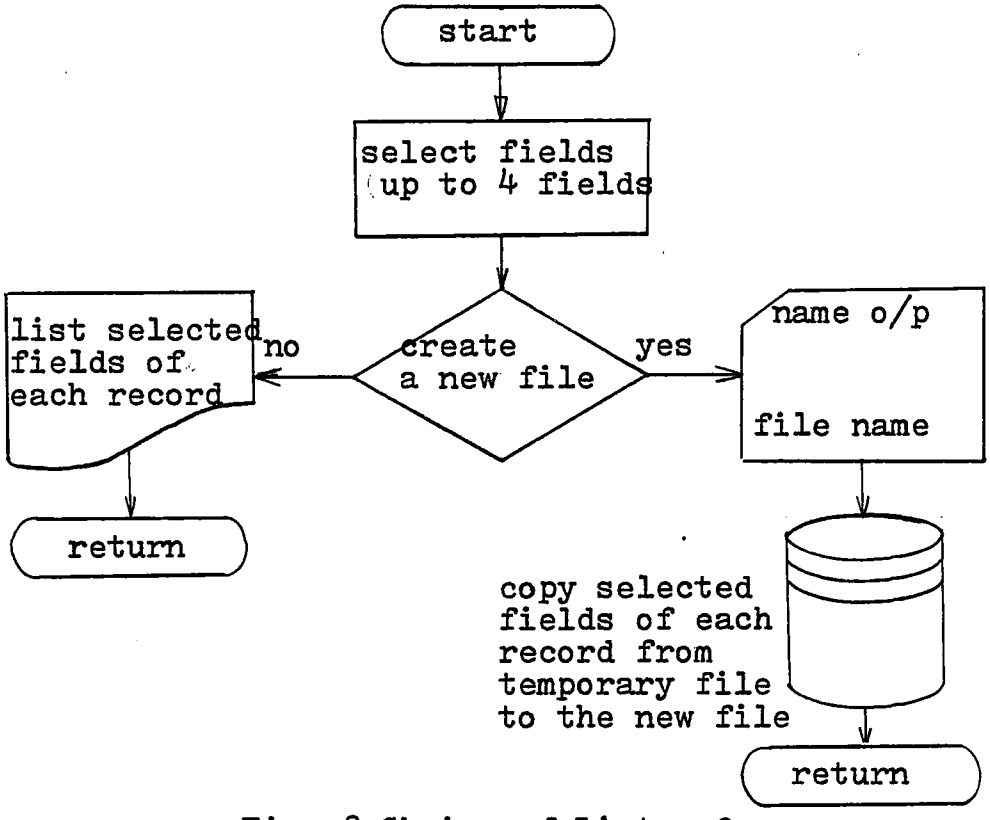


Fig. 8 Choice of List = 2

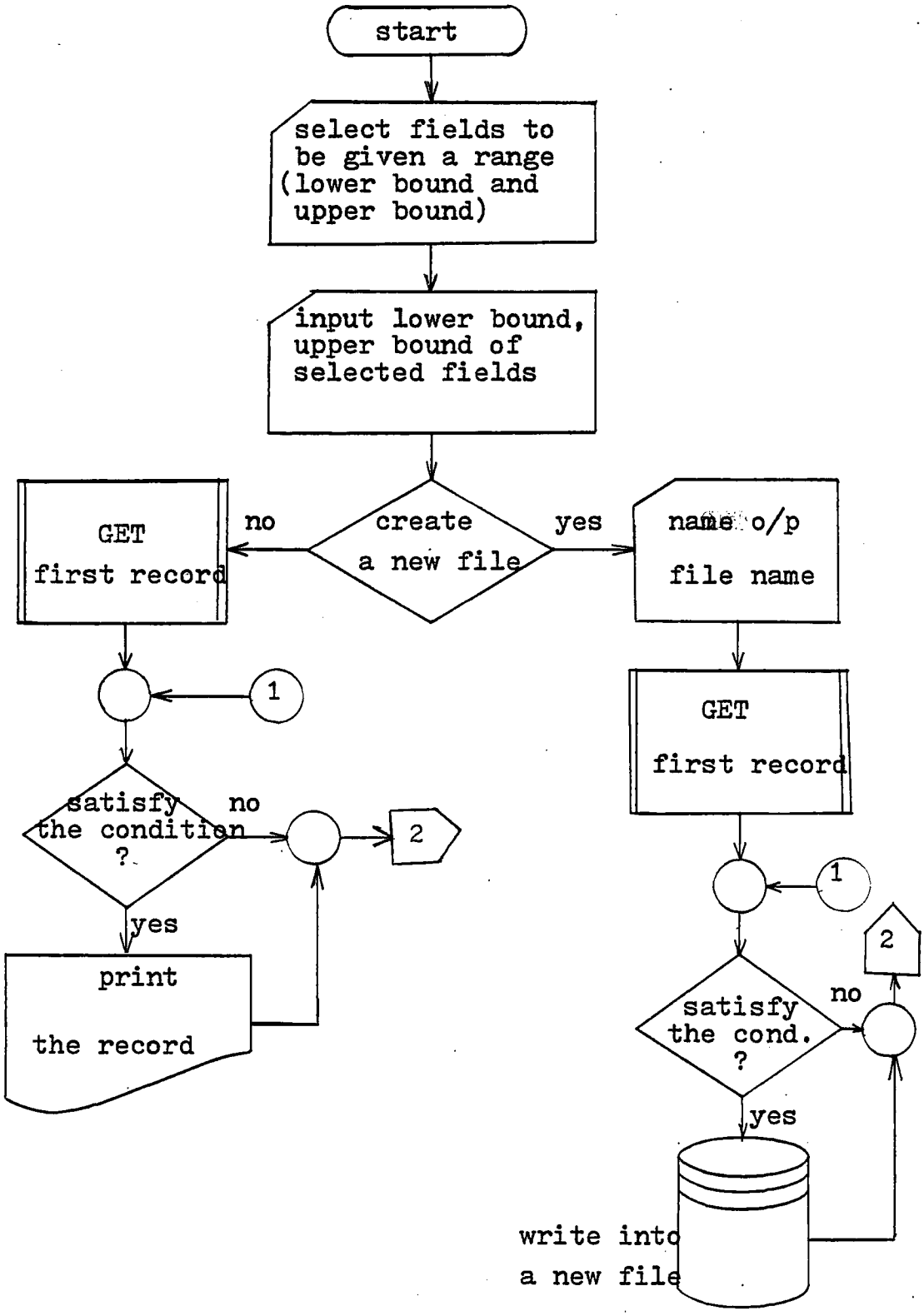


FIG. 9 Choice of List = 3

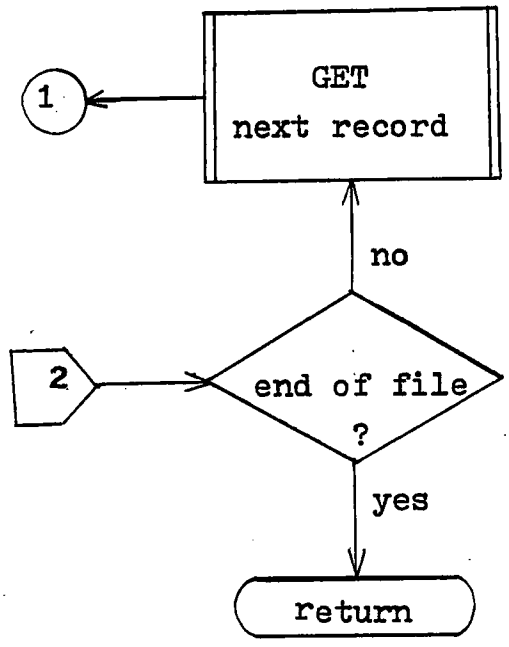


Fig.9 Choice of List = 3 (cont.)

5. Procedure READFILE

Procedure READFILE can open any file by entering file name. Each record is put into temporary file which is sorted by DATE/EV.NO.

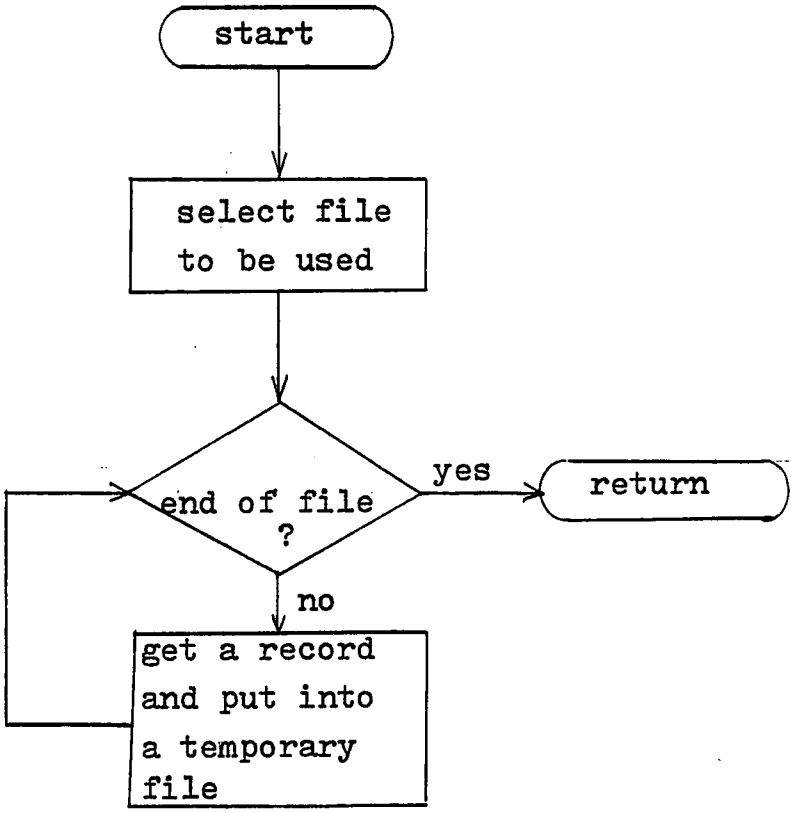


Fig. 10 READFILE flowchart

6. Procedure WRITEFILE

Procedure WRITEFILE is used for updating a file after adding a new record or deleting a record. A authority who knows a password can update a master file and a user can create his own file.

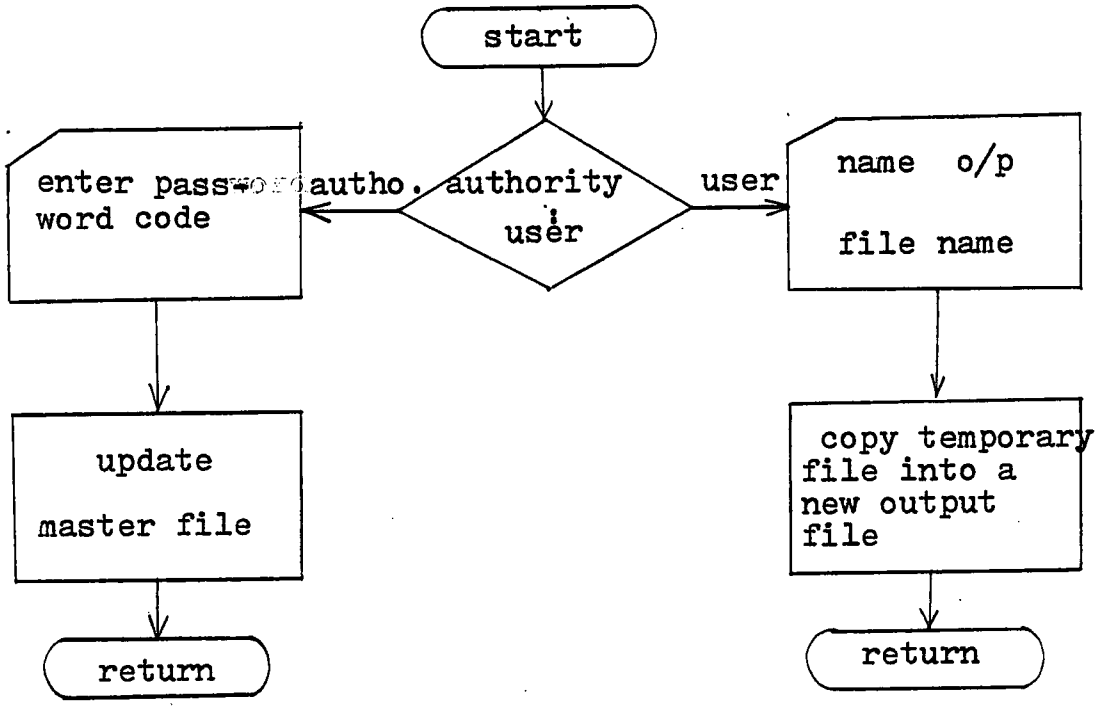


Fig. 11 WRITEFILE's Flowchart

CHAPTER III

RECOMMENDATION FOR FUTURE UPGRADES

CHAPTER III

RECOMMENDATION FOR FUTURE UPGRADES

Recommendation for Future Upgrades

In this chapter, we discuss some ideas which we would have implemented to have a more sophisticated data bases, but were not done for need of more time. Most of these would be provisions to subdivide the data bank into subfiles. This would be useful as a means of reducing the time required for searches and sorts. However, the dynamic nature of the data itself requires some of these new provisions to eliminate the need for much future program modification.

As a case in point, the record format as well as the volume of cases in current data bank is subject to future expansions, a premium must be put on the speed of record searches. This author recommends the use of a binary tree, applied to a hierarchical data model. The structure is shown in fig. 12.

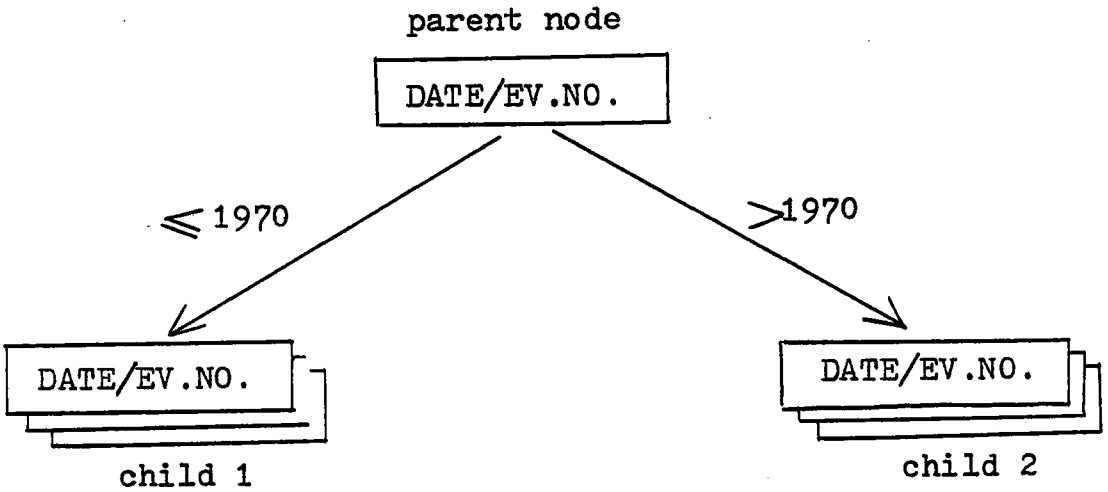


Fig. 12 Recommendation for future upgrades

The records have been sorted by year. The current range of years in the data bank is from 1960 to 1980. The parent node is the node of a decision. All events that happened in 1970 are put in the child 1 node and all events that happened after 1970 are put in child 2 node. This method will be helpful in reducing the time for searching a data file.

Another situation we would have liked to provide for is that some user groups have need only for certain subsets of the data or certain attributes of the cases. In this, we would define an access key to correspond with the attributes of interest. Following that subfiles may be created to correspond to values of that key. For example, if one is

interested in the field "EJECTION-B", one might segment the master file into three subfiles: subfile 1 contains all records with an ejection-B code in the range of 0 to 3; subfile 2, 4 to 6; subfile 3, 7 to 9. This strategy will decrease searching time.

APPENDIX A

--Glossary

--Symbol Representation

GLOSSARY

ACCESS: the operation of seeking, reading, or writing data on a storage unit.

ACCESS METHOD: a technique for moving data between a computer and its peripheral devices, eg. serial access, random access, remote access, virtual sequential access method(VSAM) hierarchical indexed sequential access method(HISAM).

ADDRESS: an identification (number,name,label) for a location in which data is stored.

BATCH PROCESSING: a single transactions are processed to completion one-at-a-time.

DIRECT ACCESS: retrieval or storage of data by a reference to its location on a volume, rather than relative to the previously retrieved or stored data.

DISK, DRUM, TAPE: a secondary storage medium.

FILE: a set of similarly constructed records.

HARDWARE: physical computer equipments such as electronic circuits, megnetic, and mechanical devices.

KEY: a data item used to identify or locate a record (or other data grouping).

MASTER FILE: a main file containing relatively permanent information that is used as a source of reference and is generally updated periodically.

ON-LINE: an on-line system is one in which the input data enter the computer directly from point of origin and/or output data are transmitted directly to where they are used. The intermediat stages such as punching data, writing tape, loading disks, or off-line printing avoided.

OPERATING SYSTEM: software which enables a computer to supervise its own operations, automatically calling in programs, routines, langage, and data, as needed for continuous throughput of different types of jobs.

PRIMARY KEY: a key which uniquely identifies a record (or other data grouping).

RECORD: a group of related fields of information treated as a unit by an application program.

SOFTWARE: a nonhardware components of the computer, in particular the programs that were needed to make the computers performed their intended tasks.

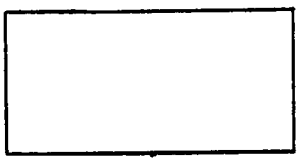
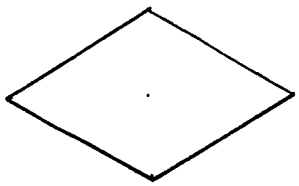
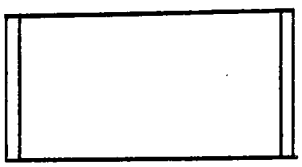
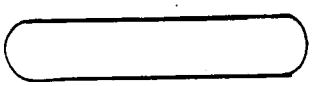
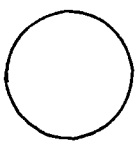
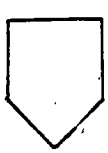
SYMBOL	REPRESENTS
	PROCESSING: a group of program instructions which perform a processing function of the program.
	DECISION: the decision function used to document points in the program where a branch to alternate paths is possible based upon variable conditions.
	PREDEFINED PROCESS: a group of operations not detailed in the particular set of flowcharts.
	TERMINAL: the beginning, end, or a point of interruption in a program
	CONNECTOR: an entry from, or an exit to, another part of the program flowchart.
	OFFPAGE CONNECTOR: a connector used instead of the connector symbol to designate entry to or exit from a page.

Fig. 13 Representation of a symbol

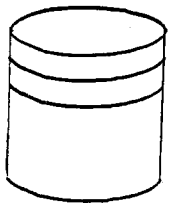
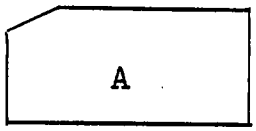
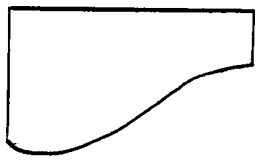
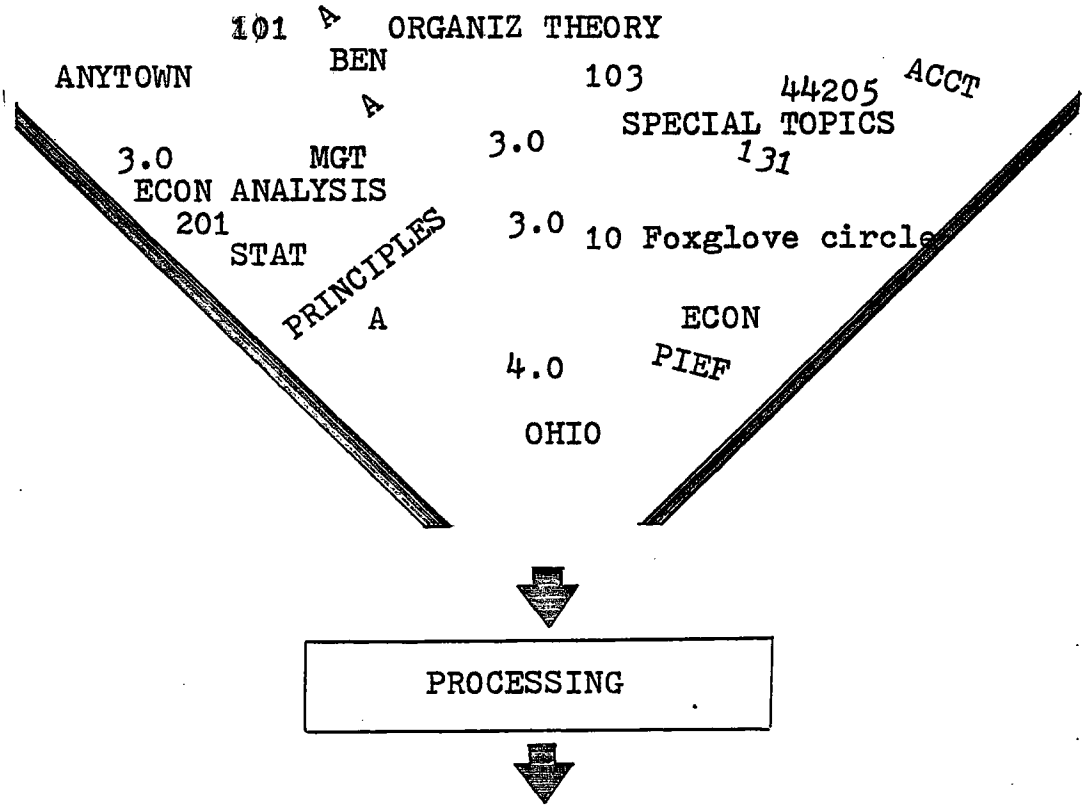
SYMBOL	REPRESENTS
	Input/output using any kind of on-line storage...magnetic tape, disk, or drum.
	Read a value from the input stream and assign it to the variable A.
	Indicates an input/output document

Fig. 13 Representation of a symbol (cont.)

APPENDIX B

-- Miscellaneous



-GRADE REPORT-

The University of Hard Knocks

Name: Ben Pief
Address: 10 Foxglove Cir.
Anytown, Ohio 44203

Debt	Course	Title	Cr	Gr
STAT	101	SPECIAL TOPICS	3.0	A
ACCT	131	PRINCIPLES	3.0	A
MGT	103	ORGANIZ THEORY	4.0	B
ECON	201	ECON ANALYSIS	3.0	A

CREDITS-13.0 POINTS-48.0 GPA-3.69

Fig. 14 Data VS. Information

Record Organization

A file (or data base) is a collection of records.
All records in a file are of the same type.
Each record has one or more fields.
Each field contains data (data item).

The above definition may describe by fig. 14.

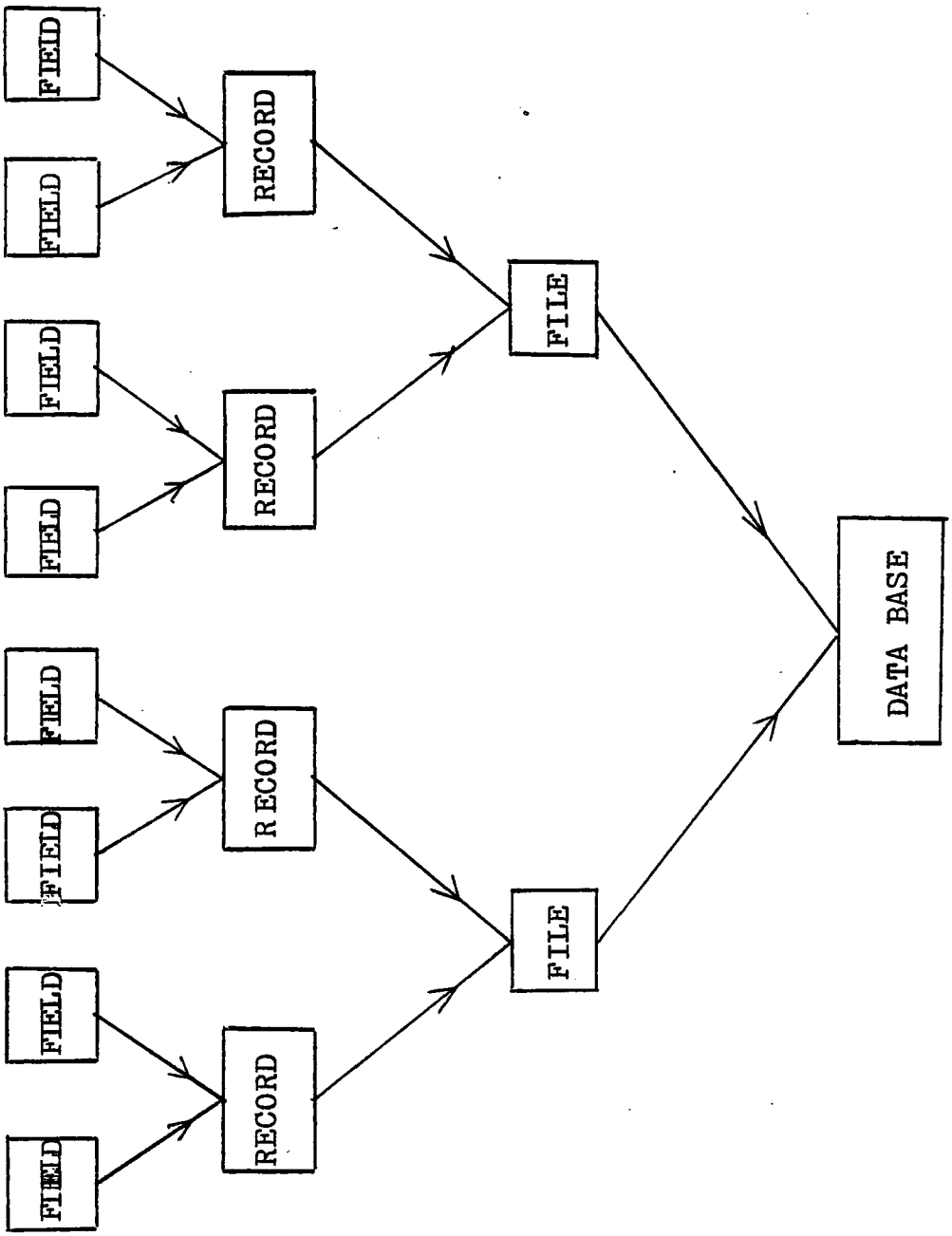


Fig. 15 Data Structure Hierarchy

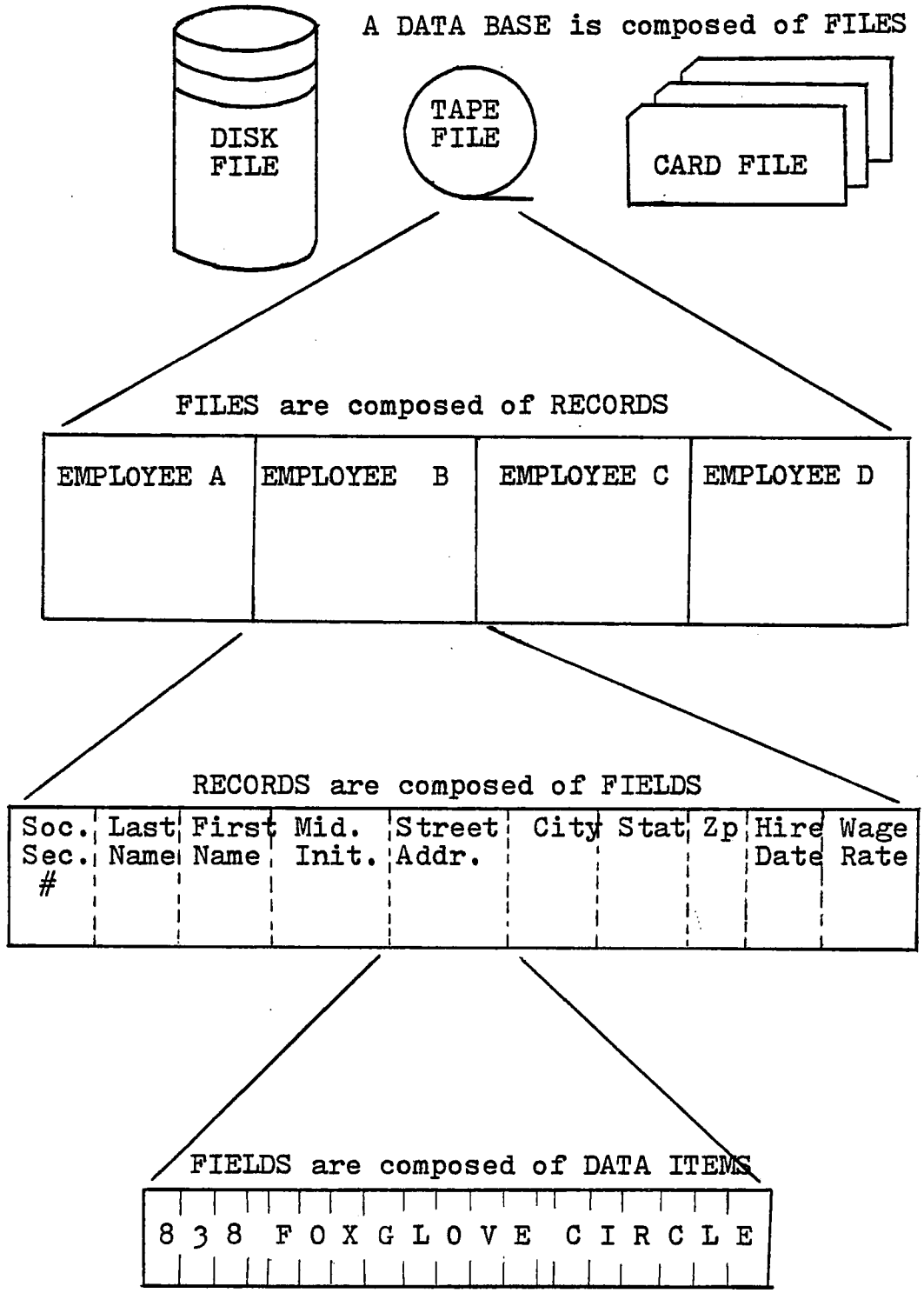


Fig. 16 The Hierarchical Nature of a Data Base

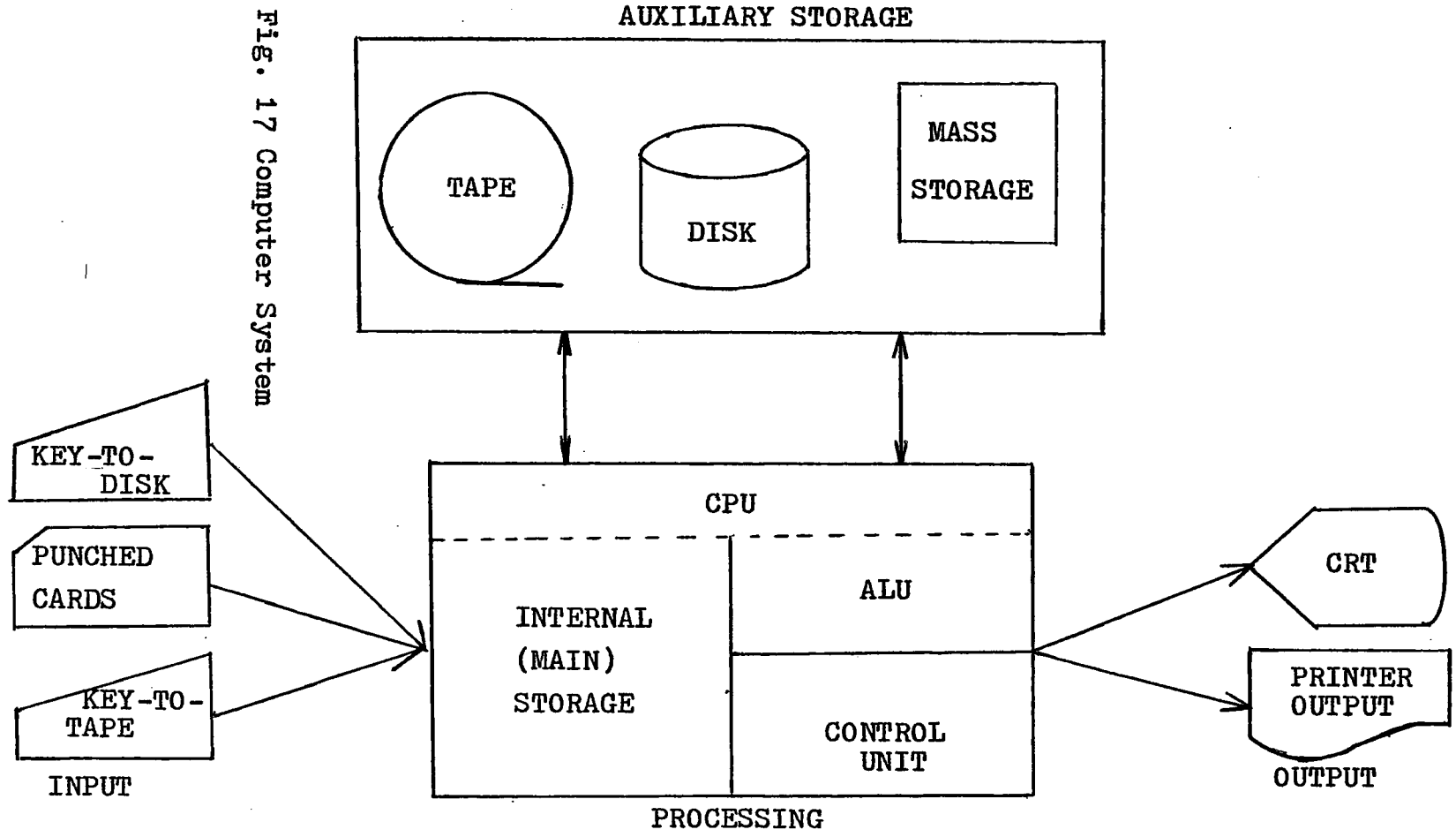
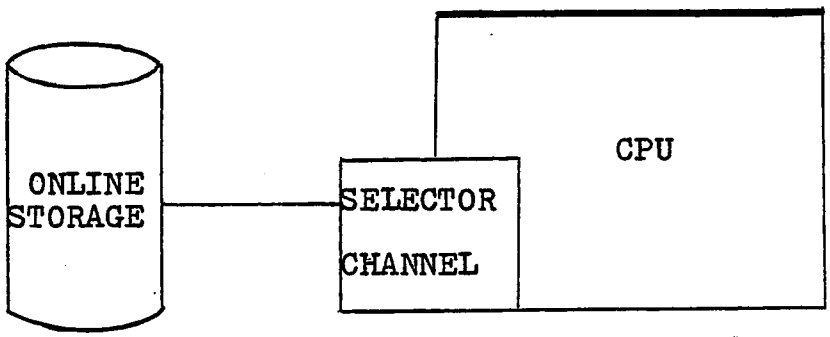
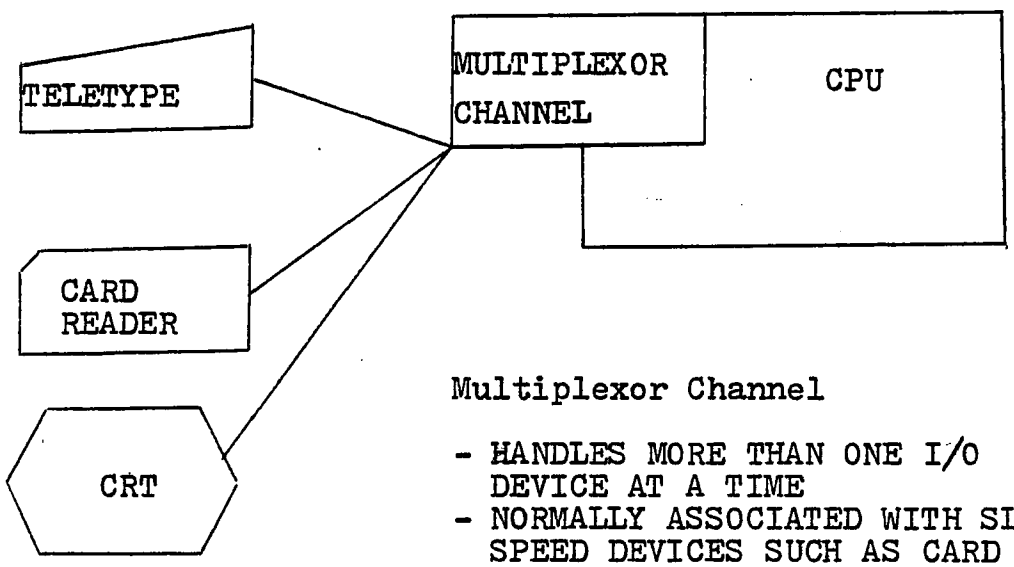


Fig. 17 Computer System



Selector Channel

- CAN ACCEPT INPUT FROM ONLY ONE DEVICE AT A TIME
- USED WITH HIGH SPEED DEVICES SUCH AS MAGNETIC DISK DRIVES



Multiplexor Channel

- HANDLES MORE THAN ONE I/O DEVICE AT A TIME
- NORMALLY ASSOCIATED WITH SLOW SPEED DEVICES SUCH AS CARD READERS, TERMINALS, AND PRINTERS

Fig. 18 Selector and Multiplexor Channels

REFERENCES:

1. Date, C. J. , An Introduction to Database System, 2nd. ed. Addison-Wesley Publishing Company, Inc., 1977.

2. Atre, s. , Data Base: Structured Techniques for Design, Performance, and Management, John W. Ley & Sons, Inc. 1980.

3. Martin, James, Computer Data-Base Organization, 2nd. ed. Prentice-Hall, Inc., Englewood Cliff, N.J. 07632 1980.

4. Ralston, Anthony , Encyclopedia of Computer Science, 1st. ed., Van Nostrand Reinhold Company, 1976.

5. Grogono, Peter, Programming in PASCAL, Revised edition, Addison-Wesley Publishing Company, Inc., 1980.

6. Nimon, William, H. , Effective Data Base Design, Prentice-Hall, Inc., 1981.