An Introduction to MINISIS

by
Donald F. Thompson
and
Mary R. Campbell

August 1979, revised July 1981

*** NOTICE ***

THIS MANUAL IN NO WAY COMMITS IDRC OR ITS AGENTS TO SUPPORT ANY OR ALL OF THE FEATURES OR FACILITIES DESCRIBED HEREIN.

THIS DOCUMENT CORRESPONDS TO RELEASE "D.01" OF MINISIS.

Table of Contents

1 Introduction

This document describes the computer based information management system developed by the International Development Research Centre: MINISIS. The system was developed to fill the need for the management of library tasks on a minicomputer. It joins the ISIS (Integrated Set of Information Systems) family of information systems, but is a new design to provide for largely interactive operation without the need for an expensive computer. Utilising the latest in hardware and software technologies, it provides the same services as its older brothers, but much more easily, and at a fraction of the cost.

Although the system was created primarily for use in a library environmment, the design is extremely general, and its use is certainly not limited to libraries. The data structure is highly flexible, and can easily be adapted to meet most requirements.

The following description is intended for a reader with some knowledge of both information systems and computers, who will be able to evaluate the potential of the system against the requirements of particular applications.

MINISIS is an information management tool designed to run only on the HP 3000 family of computers. It allows the definition and creation of data bases without resorting to any computer programming, most of the work being done in an interactive mode. The system does not require a dedicated computer; it will run on a minimum Hewlett-Packard 3000 hardware and software configuration, plus HP-KSAM.

Because the system was designed to be interactive, it must be able to function in the language of the user. It is currently available in English and French only, but each site may adapt the user interactions to the language of its choice. The system may be run in more than one language on the same machine at the same time. Spanish will be actively supported in the future.

The history of the development of MINISIS includes its use in libraries. This is an application that has many requirements for which computer-structured data has not generally been amenable. Fields of data must be of variable lengths within the same record, and within the data base. Fields may repeat many times, or may not exist at all. Fields may need to be grouped together, and these groups themselves may repeat. As a consequence of these needs, MINISIS has been designed with an extremely flexible approach to structure. This is reflected not only in the original configuration, but in the fact that a data base may be easily changed even after considerable use. Many options that affect only an individual user may be specified by him/her, relieving computer staff of the responsibility of making frequent internal changes to satisfy individual user demands.

Library applications demand that users be given rapid, easy to use access to data bases. This is provided by the QUERY processor that allows many searching operations. It is supported by a data base structure that makes such access fast and efficient. This same structure also forms the foundation for report generation processors that allow sophisticated operations to be performed on large amounts of data.

Since this is truly an interactive system, many users may simultaneously access and update the same data bases, all performing different operations. Since all users may not need access to the same data, the same information will be viewed by different people in different ways. Different data bases may be combined so that a user will see all the data as being in one place.

User access to data bases can be restricted using the security feature of MINISIS. This feature, applied by a data base manager, can limit a user's access to both data and processors.

The MINISIS system has been written specifically for the HP 3000 family of computers, under the Multi-Programming Executive (MPE). The programs are written in HP's System Programming Language (SPL), a high level language allowing access to the powerful operating system, and a sophisticated structured approach, at the same time.

2 System Design

The design of MINISIS is based on relational and domain algebra. This has resulted in a design in which problems are handled in a consistent fashion, so that the system is easily understood. Although it does not substantially affect the user, it does result internally in a system that is easier to maintain and extend. Combined with the SPL language features, and a modular programming approach, this has yielded a system that is very reliable, and yet capable of reasonably fast implementation.

A layered approach to system software was also used, so that a site wishing to write its own processor could use the system building blocks, without worrying about many of the details. This not only makes the job of the programmer easier, but also maintains the consistency of the data base, so that it is accessible by pre-existing system modules.

2.1 Why Data Base Management?

A data base may be defined as a collection of interrelated data, in this case stored in a computer. The traditional approach has been to keep all the data for one purpose stored in one place; that place is usually a file, the name given to a computer structure in which large amounts of data are stored. Since special programs are needed to access each data base, the information for one purpose is not readily available for others, and one set of information can't be related to another.

MINISIS supports the concept of an integrated data base: common information is shared, and all information is stored in a consistent fashion. Users no longer need be concerned with files - they see only their own customized version of a data base. This may represent a small amount of data from a large file, or the combination of data from many files. Because the data is structured in a controlled and consistent fashion, the same programs may be used to perform operations (e.g. sorting, editing) on any data base. The programs are generalized to function not only with different data, but different user needs.

The advantages of storing data in an integrated data base include:

- a reduction in redundancy, since the data can be stored in one place, and its relationship to other data defined
- a reduction in inconsistency, since the same data is stored only once
- standardization may be enforced, by the use of system-wide authority data bases
- data may be shared, allowing previously inaccessible relationships between data to be exploited

- data bases may be created simply and economically, since a new set of programs isn't required for each new application.

Within MINISIS, each item in the data base, and its relationship with all others, is defined. Both the characteristics and the relationships may be easily changed to meet future needs.

3 A Functional Analysis

The system is defined by the data bases and their structure, and the processors that can work on those structures.

For the user, each data base is made up of records. Each record may consist of one or more fields. A field may be repeating, a subfielded field, or a subfield:

Data base

Record

Field - repeatable
- elementary
- subfielded
- subfield

For each Record, there must be a unique identifier. In some data bases this will be an Internal Sequence Number (ISN), in others, a value (keyvalue). A record is a logical construct, not a physical one. Although the user will not be aware of it, data contained in the fields may come from many different physical locations within the computer system.

The simplest field is an elementary field. It may have one value, or many. In the latter case it is called repeatable. It may be subfielded, in which case the group of subfields may be referred to as a group, or separately. A subfielded field may be thought of as a mini-record within a record. All of its elements (subfields) belong uniquely to the subfielded field. A subfield may not repeat within a group, but a subfielded group may repeat. There may be up to 9 subfields in a group.

	Repeatable?
Elementary Field	Yes
Subfielded Field	Yes
Subfield	No

For example, if we had a data base with names and telephone numbers and areas of interest of people, it might look like:

ISN	Internal Sequence Number
Name	subfielded field (not repeatable)
last name	subfield 1
first name	subfield 2
middle initial	subfield 3
Telephone number	repeatable subfielded field
area code	subfield 1
number	subfield 2
Area of interest	repeatable elementary field

In this record there could be only one name, and the name has three parts. For each telephone number, there would be an area code and a local number. The group is repeatable, since a person might have more than one telephone. Finally, there is the area of interest, separated as many times as needed, one for each area of interest. Each repetition is referred to as an occurrence.

From the system perspective, all of this data has to go into files. Once the data base has been defined, the MINISIS system takes care of all of these details.

The functions one wishes to perform with the data base may be broken into five categories:

1) getting information into the data base
2) making sure it's correct
3) getting it out of the computer so that people can use it. This may take the form of an individual query for a few pieces of information, or the production of a large report
4) distributing data bases to other people
5) taking care of internal management of the files containing the data bases, creating and defining new data bases, etc.

For each of these areas, MINISIS has one or more processors. These are made available to each user by way of a menu. The menu the user gets depends on the level of authority given by the system manager. The system manager can define the names of the processors as desired.

3.1 Getting the Data into the Computer

New data is entered into the system using the ENTRY processor. After checking, it may be edited using MODIFY. Finally, when the record has been found correct, the RELEASE processor may be used to flag the data as available for public use.

3.1.1 The ENTRY processor

The ENTRY processor provides online data entry. Terminal operators are prompted with the name of each field indicated for automatic prompting -- they can later add any fields not normally prompted for. Operators are also given the opportunity of selecting a bibliographic level (analytic, monographic, etc.), allowing groups of fields to be prompted according to their level. Once data is entered, it is checked for the correct length; if it is numeric data, that will also be checked. Fields can be flagged for checking, in which case the contents of the field are effectively compared to all the other records in the data base, to detect duplicates. Fields may also be validated -- a process that ensures that the data entered matches the allowable data, as stored in an Authority Data Base. Authority Data Bases are maintained using the same tools as all other data bases; thus a user may enter new information into it using ENTRY, which may be invoked during the validation process.

Data is entered directly into the data base. In many cases, it will not be available to some users, until it has been released (using the RELEASE processor). Fields may be automatically inverted at the time of entry, so they will be accessible to the querying facility of the MODIFY processor.

The ENTRY processor also allows automatic entry of the current date, and the entry of diacritical codes. Entries may be made into more than one physical file in one operation.

3.1.2 The MODIFY processor

Once a record has been entered into the system, any further changes are made using the MODIFY processor. This processor is primarily intended for interactive use, and also has facilities for modifying many records, using the batch mode.

Records may be selected by their ISN, the key value (for Authority Data Bases), or by the contents of some field, using a subset of the commands of the QUERY processor. Once a record or set of records has been selected, they are individually presented to the terminal operator. If a record has been locked, it will not be accessible to MODIFY. (See RELEASE, below.)

Commands allow editing at the field or character level. Fields may be ADDed, DELeted or REPlaced. Characters within a field may be edited using the CHAnge command. Individual fields or the whole record may be displayed. The format of a record display may be selected by the terminal operator.

Throughout the whole of the editing process the integrity of the data base structure is ensured. Subfield groupings are maintained; fields flagged for validation will be verified if changed or entered again. Fields inverted online will have the inverted files updated. Data spread across more than one physical file will be updated as and where needed, and special data bases may be protected.

The GLOBAL feature in the MODIFY processor will generate a batch job, and automatically STREAM it. Record selection can be by ISN, key value or query. Then any operation may be chosen for the fields in those records selected. If fields are repeating, they may be selected by value. The GLOBAL feature frees the terminal operator from repeating the same editing function on many records, and can run on its own during off-peak machine time.

3.1.3 The RELEASE processor

The RELEASE processor allows records that have been entered into the data base to be either locked for retrieval but not for modification, unlocked for modification, or logically deleted.

When a record is entered into the system (using the ENTRY processor) it is placed in the "modify" state, and not generally available for "public" searching. Although some fields may be inverted when entered, others will not be, thus reducing the amount of overhead involved in modifying a record (since associated inverted files won't have to be changed), and restricting access to only those records checked for accuracy.

When a record has been found to be correct, it may be changed to the "retrieval" state, at which time it is available for public access, but can't be modified. In the process of changing a record's state, any fields marked for inversion on release are inverted. Should it later become necessary to modify the record, it may be returned to the modify state -- at that time any inverted-on-release fields will be "uninverted".

Records may be logically deleted from the data base using the delete command. They must first have been enabled for modification. Records will not be physically removed, since more than one data base may access the same physical record. If all data bases associated with the record have deleted it, the GARBAGE utility will eventually physically remove it.

Records for processing by RELEASE may be chosen by ISN or key value, or by a hitfile produced by the QUERY processor, thus allowing values in the record to determine whether its status is changed.

3.1.4 The PRINT processor

The PRINT processor forms the heart of all record displays within MINISIS. It allows the user to determine the exact form of output. Fields may be preceded or followed by fixed textual material (literals), their spacing and length may be controlled, and the page position may be controlled. Output may be in tabular or sequential form, or may be set up for preprinted forms.

The PRINT processor will display data bases themselves or combine the output from INDEX, QUERY or COMPUTE with their originating data base, allowing the original records to be ordered for printing, and optionally combined with results from INDEX or COMPUTE. The operation of these processors, combined with the flexibility of output formatting provided through PRINT, allows the creation of sophisticated reports.

New print formats are generated and existing formats are edited using PRINT, following an interactive dialogue. The user need know no special codes, and has but to provide answers for questions concerning the various options. This allows users the freedom they want, without reliance on the special knowledge of a "computer person"; it provides not only less work for highly trained computer personnel, but also a happier user community.

PRINT will output to a user's terminal, the system line printer, or to a special device if special forms are used. Paging characteristics, including the width and depth of each page, record splitting, number of records per page, printing of diacriticals, page headings, etc. are controlled by the user as part of the print formatting process.

3.2 Accessing the Data

Once the data has been entered and checked for accuracy, ease of access becomes the focus of interest. The heart of the access methods lies in QUERY, which allows records to be selected based on the content of specified fields within the record. It allows searchers to specify the contents of the records desired, and then have them printed at a terminal, or on the system line printer, or the list of records may be saved for further processing by INDEX or COMPUTE, and then finally output by PRINT. The INDEX processor performs various sorting operations, and accepts as input the hitfiles from QUERY, simple ISN ranges, or output from INDEX itself.

3.2.1 The QUERY Processor

This processor allows records to be retrieved interactively, according to the contents of various fields. It functions either by examining each record (a relatively slow, but quite precise method) or by accessing inverted files (a much quicker, but somewhat less precise method). Inverted files may also be referred to herein as "fast access" files.

Inverted files are created by extracting keys from fields, and making a list of records which contain each key. Keys may be inverted in a controlled or uncontrolled fashion. In the uncontrolled method, keys are generally associated with words. Thus the title

```
     The Rise and Fall of the Roman Empire
      1    2    3    4   5  6    7      8
```

would yield 7 keys; "the" is repeatable, and will only be considered once. The system will allow inversion of all of these keys. It is frequent practice to remove "noisewords" such as "the", "and", "of" and the like, since they appear so frequently, and this is supported by allowing the user to maintain stopword files, which will comb out all of these keys before inversion.

In controlled key inversion, each key is checked against a data base of legal keys, and the inversion is carried out only if the key extracted is indeed allowable. This method assumes that the person originally indexing the item will be able to select a keyword which will also occur to the searcher. It has the advantage that different forms of the same descriptor won't appear in the data base, and thus cause items to be missed by the searcher who doesn't think of all the variant forms.

A key can be defined in various ways. It may occur between user-defined special characters, or be delimited by spaces or punctuation or the beginning of the field.

A sample record might be:

```
Figueroa, J.J.
     Society, schools and progress in the West Indies.
     Oxford, Pergamon Press, 1971. 208 p.
        370(729)     F 4

Monograph on /education/ in the /Caribbean/ (West Indies) - discusses
the /history/cal background, /economic conditions/, /social conditions/,
present /school/s and /educational system/s of the area, /educational
need/s, /educational planning/ and /teacher training/, etc.
/Statistical data/, /bibliography/.    7124
```

This might yield two sets of keys: those in the title, and those in the abstract. The title field, under normal circumstances, would yield the keys

SOCIETY
SCHOOLS
PROGRESS
WEST
INDIES

while the abstract would yield all those keys between slashes. For the purposes of our example, we will assume all of the keys in the abstract field to be in the thesaurus (see below for details).

To access this record, the searcher would log on to the system, and automatically be provided with a menu of available processors. Having selected QUERY she would be asked for the name of the data base, and would then ready to perform a search. A HELP command will provide a list of valid commands, should she forget.

The core commands in the QUERY processor are the Boolean operators: AND, OR, EOR, NOT. With them, the searcher can combine lists of records containing various keys. Various other utility commands allow the selected records to be viewed, or saved for further processing, listed offline on the high speed printer, etc. In addition, there are Thesaurus Operators, discussed below.

The AND, OR and EOR operators join two lists of records; NOT works on only one list. Each time a key is entered during the search, a list of records containing that key is generated and saved in a temporary area. The lists may be manipulated using these operators.

The AND operator joins two lists by specifying a smaller set of records -- those contained in both lists.

The OR operator specifies a wider set of records -- those contained in either of the two lists, or both.

The EOR operator (Exclusive OR) specifies a subset of the OR operation -- those records belonging to one list or the other, but not to both.

The NOT operator specifies all those records not in the list.

By way of example, we will formulate a search for the record given above. Because more than one search may be performed during a run of the search processor, it must start with an equal sign (=) and end with a dollar sign ($). The equal sign should be at the beginning of the first line of the search. In our search, we will use the translation facility, which translates some descriptors into other languages; in this case from English into French and Spanish. (This feature will be discussed later: see the Multilingual Thesaurus.)

The search might look like this:

```
     > = education
```

will initiate the search and look for the key "education", in the default search field -- in our case, the abstract field. The processor will reply with

```
        EDUCATION     P=795
        EDUCACION     P=10
              1:   P=805    T=805
```

The key is the same in English and French, so we only see one translation: into Spanish. The P= after each key indicates the number of postings: records that contain this key. The lists of records containing either language version of "education" are all OR'ed together, so that a record containing "education" or "educacion" or both, will be in the list. The final line shows the number of postings, and the total to date (T=805).

Since 805 records is rather too many to look at, we will specify that it also has to do with schools:

```
     Q> and school                 | user input
        SCHOOL    P=51             | English
        ECOLE     P=3              | French
        ESCUELA   P=1              | Spanish
               2:   P=29    T=29
```

This will cause a list of all the records containing "school" and its translations to be generated -- 55 records in all. The "and" in our input says to perform the AND operation on the records selected in this command, and those in the one immediately preceding it, so records must contain both "education" and "school". The T=29 tells us that 29 records contain both keys.

As we wish to restrict the search further, we might search for a key from the title:

```
     Q> and titlem society
        SOCIETY        P=62
               3:   P=2     T=2
```

The "and" means the same thing it did above. "Titlem" is the mnemonic for the monographic title field. "Society" is the word in the title we want to see.

From the T=2 we can see that we have only 2 records in our list. Invoking the BROWSE command will display those two records, in any format specified by the user, or in the default format supplied with each data base.

The whole search could have been performed in one line:

```
   > = education and school and titlem society
```

All of the logical operators may be combined in this way, and parentheses may be used to assure the correct searching order.

Had we known only that the book was by Figueroa, and about education, we might have entered

```
              > = education
            Q>  and pauthm Figueroa
```

Since the author field in this sample data base isn't inverted, this would have caused QUERY to execute a free text scan: each record containing "education" or its translations would be read, and the field PAUTHM (Personal Author-Monograph) would be extracted, upshifted, and compared with the text string FIGUEROA. When the text matched, a "hit" would be noted, and a record kept of that record number. Since this can be a lengthy process if many records are involved, the procedure may be prematurely halted at the user's option. A free text scan of inverted fields can also be carried out if the user so requests.

Once a set of records has been chosen with the QUERY commands, all or part of it may be listed at the terminal or on the system line printer, in interactive or batch mode. The list may be saved for future processing by COMPUTE, INDEX or PRINT. The search formulation, or parts of it, may be saved for use on other data bases, and the data base itself may be changed. Should assistance be required, the HELP command will provide a list of legal commands at any time.

A limited range of mathematical operations (+, -, /, *, parentheses) may be performed on all the records selected in the last query line, with the result shown on the terminal.

Searching can take place using the right truncation feature, where the user supplies a character string, and those keys which begin with the character string are OR'ed to produce a hitfile.

Another feature of QUERY is DISPLAY, which allows the user to browse through an inverted file online, starting either at the beginning of the file or at any key in the file, and ending at any point in the file specified by the user. Both keys and their postings will be listed. If the inverted file has a thesaurus structure (see 3.2.1.1 below), the other-language versions of each key will be displayed with the key, as well as its related, broader, narrower and ANY terms.

3.2.1.1 The Multilingual Thesaurus

In the above example, most of the keys searched for were thesaurus terms. A thesaurus is a terminological control device used to describe an item in the data base. The thesaurus specifies a descriptor for each concept, and relates the concepts in an hierarchical manner. It also specifies the context in which a concept should be used to describe an item. A thesaurus is multilingual when a concept is assigned a descriptor in more than one language.

From the perspective of a person describing an item when entering it into a data base, the thesaurus represents a list of controlled-use descriptors that aids in translating natural language into a more restrained system language. The searcher must also make this translation, and can also take advantage of the hierarchical structure, which is represented in a special MINISIS data base, accessible through Thesaurus Operators.

For a given descriptor, there may be broader terms (BT), narrower terms (NT) and related terms (RT). These are related by a hierarchical structure. The broader term is less specific (and thus lower in the hierarchy), the narrower term more specific (and thus higher in the hierarchy), and the related term is at the same level of specificity, but in a related area (thus at the same level in the hierarchy).

By way of example, we might have the term

SCHOOL

it has a broader term:

EDUCATIONAL INSTITUTION

and several narrower terms:

COMMERCIAL SCHOOL
COMPREHENSIVE SCHOOL
EXPERIMENTAL SCHOOL
MOBILE SCHOOL
NURSERY SCHOOL
PRIMARY SCHOOL
SECONDARY SCHOOL
TECHNICAL SCHOOL
VOCATIONAL SCHOOL

and a related term:

SCHOOL SYSTEM

Within QUERY, one simply uses the BT, RT, NT operators before a thesaurus term. Thus

BT SCHOOL

would retrieve all of the records containing school, as well as its broader term EDUCATIONAL INSTITUTION. The RT and NT operators work in the same way.

The system also supports the use of ANY tables. These are not formally part of the thesaurus structure, but instead are more of a searching aid, used to refer to groups of thesaurus terms that will frequently be used together. An example would be the identification of a geographical area:

```
                 FAR EAST

Any Terms:       CHINA
                 HONG KONG
                 JAPAN
                 KOREA DPR
                 KOREA R
                 MACAO
                 MONGOLIA PR
                 TAIWAN
```

All of the ANY terms, plus the name of the ANY group, will be logically OR'd together, resulting in a list of all the records containing those terms. Both the ANY terms and the name of the group must be valid descriptors.

Within QUERY, the ANY operator is used:

```
      > = ANY FAR EAST
          FAR EAST              P= 45
          CHINA                 P= 295
          HONG KONG             P= 62
          JAPAN                 P= 203
          KOREA DPR             P= 12
          KOREA R               P= 165
          MACAO                 P= 2
          MONGOLIA PR           P= 3
          TAIWAN                P= 93
               2:    P=801      T=801
```

Underlying all of this structure is the fact that the thesaurus is multilingual -- it supports up to ten languages, and will perform translation between terms of different languages representing the same concept. Within QUERY the languages on which to search may be specified -- it defaults to all languages.

When the translation feature is enabled, terms may be entered in any language. The system will automatically generate the other language versions for that term, display them with the number of records containing them, and then keep a list of all the records containing any of the terms (that is, a logical OR). If a thesaurus operator is used, the translation will also be performed, so that all of the languages for all of the terms accessed will be presented. Continuing with our example above:

```
> = SCHOOL
  SCHOOL     P=53
  ECOLE      P=3
  ESCUELA    P=1
         1:    P=57     T=57

> = BT SCHOOL
  SCHOOL    P=53
  ECOLE     P=3
  ESCUELA   P=1
  EDUCATIONAL INSTITUTION         P=79      | BT in English
  ESTABLISSEMENT D'ENSEIGNEMENT   P=3       | BT in French
  ESTABLECIMIENTO DE ENSENANZA    P=2       | BT in Spanish
         2:    P=138    T=138
```

The thesaurus structure also supports the use of Forbidden Terms. These are terms that have been singled out as being illegal for use as a descriptor, and have associated with them a Use Term -- the correct term to be substituted. For example

EDUCATIONAL GUIDANCE

might be illegal, and in its stead

SCHOOL GUIDANCE

should be used. Within QUERY, one may enable the forbidden option; i.e. should a forbidden term be entered, the correct term will be substituted, and the records for the correct term selected. If translation is enabled, it will be performed on the correct term.

For further information on thesaurus structures, the reader may see:

Lancaster, F.W.: Vocabulary Control for Information Retrieval.
Information Resources Press, Washington, D.C., 1972

Gilchrist, A.: Thesaurus in Retrieval
Aslib, London, 1972

Soevgel, D.: Indexing Languages and Thesaurus: Construction and Maintenance.
Melville, Los Angeles, 1974

Aitchinson, J., Gilchrist, A.: Thesaurus Construction: A Practical Manual.
Aslib, London, 1972

3.2.2 The INDEX processor

All sorting operations within MINISIS are performed using INDEX. Its basic operation is to allow the extraction of data from records; then sort it, using up to five sort keys; and finally to output it for further processing, or display using PRINT.

Data is extracted for each key to be sorted. More than one field may be associated with a key, and alternatives may be specified where fields don't exist. For every key, various kinds of preprocessing may be specified: terms extracted may be checked against a validation file, or checked for their presence in a noiseword list. Special processing may be handled by specifying a user-written routine to operate on the field. Sorting may be in ascending or descending order, and may be done using the normal ASCII sorting sequence, or Spanish, numeric or UDC (Universal Decimal Classification) order.

At the field level a number of further options are allowed: fields may be broken into terms, words or phrases, or the entire field may be extracted. As well, extraction will be performed for KWIC indexes. Various characters may be stripped from the field before sorting: non-alphabetical, non-alphanumerical, or diacritical characters may be specified, or the user may supply a list peculiar to the application. Prefixing and suffixing may be specified, as well as the number of occurrences of a subfield to extract, or keys to generate.

Once the processed data has been sorted, it is copied to an output file. At this time the original or processed data may be chosen.

Every record output by INDEX from a master data base contains the ISN of the record from which the data was taken; authority data bases generate the key value. In this way they may be joined to the original record in the PRINT processor, allowing the master records to be ordered by various fields, without moving all of the data.

Index Processing Sequence

```
field extraction
          |
          |
          V
pre-sort processing
          |
          |
          V
```

```
sorting
   |
   |
   V
output
```

3.2.3 The COMPUTE processor

This processor allows operations to be performed both within a record, and across a series of records. Within a record, fields may be operated on using addition, subtraction, division and multiplication. Sums, averages, minima and maxima may be generated across a series of records.

As input, COMPUTE will accept a hitfile, or an INDEX or COMPUTE output file; this data may then be used in conjunction with fields from the data base from which the input list was first extracted.

Various "program control" options are open to the user: computations may be conditional on the value of a given field, subtotals may be evaluated when a field value changes, etc. The statements provided to the processor are English-like, and may be entered directly, or from an Editor file. If the same statements are to be used repeatedly, the processor will save them for further use.

Once the specifications have been entered, COMPUTE will execute interactively or will stream its own batch job.

The output is similar to an INDEX output file, and is suitable for printing using PRINT. Within PRINT it may be used in conjunction with the Master file from which the data was extracted. If an INDEX file was used as input, fields may be copied to the COMPUTE output, if needed for printing. Up to 99 output fields may be created. Those containing numbers will be of the numeric type, with an accuracy of up to 20 digits.

3.2.4 The PRINT processor

This processor was discussed above. It forms part of the system at several levels: in entry and modification, it is used for prooflists, and to determine output formatting at the terminal. Within QUERY, it controls output at the terminal when browsing through a data base, as well as output on the system line printer. For processing reports, it will accept output from the INDEX and COMPUTE processors, and print it along with data from the data base used to generate that file.

3.2.5 Data Distribution (ISOCONV)

Having developed a data base, it is sometimes desirable to distribute it to other locations. Although this may of course be done by sending copies of the MINISIS system files, such a method won't always allow the needed flexibility -- either because the receiving site is running a system other than MINISIS, or because they don't want the information structured in exactly the same manner.

To facilitate data exchanges, MINISIS has adopted the ISO 2709 format for bibliographic interchange on magnetic tape. The ISOCONV processor allows such tapes to be generated or read, with a translation performed between ISO tags and MINISIS field tags. The information on the tape may be either EBCDIC or ASCII character codes.

The system manager can use ISOCONV to restructure existing data bases.

3.3 Data Base Maintenance

The system manager needs a set of tools that go beyond those of the user. He/she must define data bases and their inter-relationships and allocate space in which those data bases will reside. If necessary, user access to data bases must be restricted. The DATADEF processor provides all of these services. When data bases are in use, certain maintenance operations must be periodically performed, such as collection of unused space (GARBAGE), and regeneration of inverted files (INVERT).

3.3.1 DATADEF

This processor allows the specifications for a data base to be generated or edited. Each field is specified in an interactive dialogue. The physical files wherein the data resides are built and initialized to user supplied parameters. New data views may be created by copying and editing an existing definition. All system files that cannot be simply accessed using Editor are maintained in a consistent fashion.

This processor also contains the security feature which enables the data base manager to restrict user access to data. This is done at two levels - the data bases which a specific user may access are defined, and the processors that the user may apply to these data bases (i.e. ENTRY, MODIFY, RELEASE) are also defined.

3.3.2 GARBAGE

This function allows the data base files to be scanned for unused space, since new or modified records can't always be entered in the same place. A new version of the file, in the most compact form, is created. File sizes may be changed, and postings in inverted authority data bases may be reset.

3.3.3 INVERT

This utility processor allows the off-line generation and maintenance of inverted files. Keys are first extracted from the data base using any of the options of INDEX, and then lists of postings are created and stored by these processors. The inverted files are stored in either B-tree files or authority data bases (KSAM files).

3.4 Utility Programs

A set of utility programs for user and system manager use are also accessible within MINISIS. They include:

LISTFORMAT - allows the specifications contained in a print format file to be listed on the system line printer

LISTDDT - will list the data definition of a data base, including the name, mnemonic, tag and length of each field, on the system line printer

RENUM - allows a data base to be "renumbered"

THLOADER - is the utility processor for loading the multilingual thesaurus, and its structure terms

KWLOADER - allows all keywords in the system, as used in each processor, to be changed

RECOVERY - if MPE transaction procedures are found, allows recovery of lost data from log files after system failure

4 Support Services

The MINISIS support group is dedicated to the support and development of the system. The software is licensed under a purchase/maintenance agreement, including an original one-time purchase price, and an annual support fee.

The purchase price entitles the user to installation and user training.

The support fee entitles the user to periodic updates and improvements, as well as a limited amount of consultation regarding special problems.

Purchase of the system also entitles the user to membership in the MINISIS Users' Group, which meets to discuss common problems and discuss the nature of future enhancements to the system.

5 Data Structures Within MINISIS

Basic element: field

- identified by tag or mnemonic

  tag: Annn or Ann.n, where A is a letter A to Z (excluding Y)
  n is any digit
  eg.: A120, Q429, Q42.9

  mnemonic: any combination of up to 6 alphanumeric characters

- may be subfielded, with up to 9 subfields in each group. Group field name must have last digit 0. Subfields may not repeat within the group, the letter and first two digits within their tag must match the group field name, and the last digit, which must be 109, indicates their position in the group.

```
eg.:  A190        group field name
      A191        -
      A192        |
      A193        + - subfields
        .         |
        .         |
        .         |
      A199        -
```

- non-subfielded fields may have any tag ending in zero
- non-subfielded fields and subfielded groups may repeat up to 60 times
- contents may be any character but exclamation mark (!)
- length of each field is limited only by record size

Fields make up: Record

- identified by Internal Sequence Number (Master format), or key value (Authority File)
- may have up to 256 fields defined
- maximum length of master file record: 4096 characters including directory space (8 characters per field occurring) plus leader of 12 bytes
- maximum length of authority file record: 2042 characters

Records make up: Data base

- may consist of from 1 to 18 physical files
- greater than 1 million records
- structure may be changed easily after data entered

6 User Processor Summary

ENTRY

- interactive data entry
- automatic field prompting, with selection of up to four sets of fields to be prompted (bibliographic levels)
- fields checked for correct length, and content if numeric
- may check for occurrence of value of field in data base
- may validate value against allowable values in another data base
- data entered directly into data base, may become available immediately
- inverted file may be created and updated at time of data entry
- automatic numbering of new records

MODIFY

- for interactive or batch modification of records
- only allows modification if record has not been locked by RELEASE
- will update inverted files, and do validations

RELEASE

- marks records as MODIFYable, or only retrievable (read only)
- records may be in different states for different data bases
- also performs logical deletions
- runs in interactive or batch mode

QUERY

- interactive or batch searching processor
- easy-to-use commands
- Boolean operators AND, OR, EOR and NOT may be used to combine lists of selected records
- records may be selected using inverted files or by comparing text
- lists of records may be saved for listings offline or further processing (INDEX, PRINT, COMPUTE), or records can be displayed on terminal
- supports use of multilingual thesaurus, with up to 10 languages
- thesaurus operators allow selection of Related, Narrower or Broader terms
- frequently used thesaurus terms may be kept in ANY tables
- user can browse through inverted files, including thesaurus
- searches can be right-truncated

INDEX

- sorts records, using up to five sort keys
- various methods of selecting fields for use as keys, and for preprocessing before sorting
- records to be sorted may be chosen by ISN or key value, or using record lists from QUERY
- output may be PRINTed, or further processed by COMPUTE
- runs in interactive or batch mode

COMPUTE

- allows arithmetic processing of selected fields
- takes input from INDEX output, or QUERY record lists
- arithmetic operations within records, sums, averages and max/min across groups of records
- will create break subtotals
- accuracy of 20 digits
- output may be printed by PRINT, or further processed by INDEX or COMPUTE
- runs in interactive or batch mode

PRINT
- generalized display processor
- allows user-defined generation of print formats
- will print records directly from data base, or combined with output from INDEX or COMPUTE, or using QUERY record lists
- control formats on terminals or system printer
- allows generation of diacritical characters
- runs in interactive or batch mode

ISOCONV
- allows generation or reading of tapes in ISO 2709 international data exchange format
- converts external data base to MINISIS internal structure
- runs in interactive or batch mode

Appendix A

HP 3000 Computer Systems

The HP 3000 is a general purpose data processing computer system which can simultaneously perform time sharing (on up to 64 terminals), batch and transaction processing operations. The computer incorporates hardware features such as stack architecture, variable-length code segmentation, virtual memory, program protection, and dynamic file storage allocation. Hewlett-Packard offers a wide range of user-oriented software products including:

1. A powerful disc-based operating system (MPE) which is common to all members of the HP 3000 family;
2. High-level programming languages such as:
   - i) COBOL
   - ii) FORTRAN
   - iii) APL
   - iv) BASIC
   - v) RPG
   - vi) SPL (HP's System Programming Language);
3. Data base management facilities;
4. Sequential, keyed sequential, and random file access methods;
5. Data communications facilities;
6. An easy to use text editor (EDITOR)
7. A general file copying utility
8. A facility for ordering records in a file and merging files.

Hardware

Specific hardware differs for the Series 33, Series II, and Series III. In general, however, the HP 3000 system hardware includes the central processing unit, main memory, and the various peripheral devices that are available for each series.

Some of the hardware features incorporated in the HP 3000 family are:

1. Stack Architecture

   A hardware stack is used for the execution of most instructions rather than general registers, in order to provide dynamic, private, hardware protected data storage for each user.

2. Virtual Memory

   The MPE operating system uses both main memory and a dis storage area to provide a total memory space that exceeds the main memory size of the computer system. Programs and data are subdivided into units that are dynamically moved by MPE into main memory for execution. This allows programs which are larger than main memory to be compiled and executed. As a result of virtual memory, many large programs may be executed concurrently with one another and the operating system.

3. Concurrent I/O and CPU Operations
   Many input/output operations can be performed concurrently with CPU operations. The hardware enabling this operates under control of the MPE operating system which handles all queuing and device scheduling.

4. Input/Output Conveniences
   MPE treats all I/O devices as files or groups of files, which allows the user to access the devices by file names rather than by device types or logical unit numbers. The file names specified in programs are independent of the devices used for input and output, and need be associated with these devices only at the time the programs are run, either in batch or time sharing mode.

5. Security
   Each user operates in an environment protected from interference by others. System access is controlled by an account/group/user structure with optional passwords at each level. Program protection, while executing, is provided by the hardware. File security is based upon a series of passwords and hierarchical access restrictions that allow the user to specify the degree of security desired for each file created by the user.

6. Fault Control Memory
   The HP 3000 uses high speed semiconductor memory modules which provide automatic fault detection and single-bit correction.

Operating System

The Multiprogramming Executive operating System (MPE) is a disc-based software system which supervises the processing of all user programs on the HP 3000. MPE allows multiple users, running in time sharing or batch mode, to concurrently access the computer system resources. MPE allocates such resources as main memory, the central processing unit, and peripheral devices to each program as needed, as well as coordinating all user interaction with the system.

MPE simultaneously permits and controls interactive program development and execution, batch program execution, data inquiry, data base updates, and data communications. The operating system monitors and controls program input, compilation, run preparation, loading, execution and output. It also controls the order in which programs are executed and maintains usage records of the hardware and software resources they require.

Other features of the operating system include:

1. A file management system with file backup and security;
2. A friendly, powerful command language, including user-defined commands, conditional job control, and on-line HELP facility;
3. Device and file independence;
4. Complete, automatic terminal management for both local and remote asynchronous and synchronous devices;
5. Spooling of input and output;
6. Private disc volumes;
7. Power fail/auto restart.

All input and output to peripherals is handled by the MPE I/O system. It receives I/O requests from other system software, queries them if necessary and performs the transfer of data to or from the device.

A more complete description of all HP 3000 hardware and software products is available in HP 3000 Computer Systems-General Information Manual. This document is available from any Hewlett-Packard sales office.

MINISIS Installation Requirements

1 Minimum hardware requirements

- Hewlett-Packard 3000 Series II, III, 33 or 30
- minimum 256K bytes memory
- disc storage
- 1600 bpi tape drive
- terminals

Most sites will find a high speed printer, with upper and lower case, necessary for generating hard-copy output.

2 Minimum software requirements

- MPE III
- HP FOS (Fundamental Operating System: includes EDITOR, FCOPY, SORT/MERGE)
- KSAM
- SPL if program development will be done

2.1 Knowledge of MPE commands and programs

Sites will require personnel with the following knowledge of the system:

- one person who has taken the HP System Manager's Course, or who has equivalent knowledge

- one person who has taken the HP Comprehensive Introduction Course, or has equivalent knowledge, and who will act as a data base manager

- people with a knowledge of the HP publication "Using the HP 3000", who will be able to use the advanced processors

- people with minimal knowledge of computers, who will perform data entry, etc.

3 Operating system minimum configuration requirements

- MINIMUM CODE SEGMENT SIZE - 10240 words

- MINIMUM STACK SIZE - 31232 words

- MINIMUM EXTRA DATA SEGMENT SIZE - 12288 bytes

- MINIMUM NUMBER OF EXTRA DATA SEG/PROCESS - 6

Further information on system configuration will be found in the HP System Manager's Reference Manual.