DOCUMENT RESUME

ED 074 759 EM 010 947

AUTHOR Embry, Jonathan D.; And Others
TITLE GANDALF: A General Alpha-Numeric Direct Access Library Facility.
PUB DATE Oct 72
NOTE 16p.; Paper presented at the Rio Grande Chapter of the Association for Computing Machinery (October 13, 1972)

EDRS PRICE MF-$0.65 HC-$3.29
DESCRIPTORS *Computer Programs; Computers; *Data Bases; Information Processing; *Information Retrieval; *Information Systems; Library Research; Search Strategies
IDENTIFIERS ERIC; GANDALF; *General Alpha Numeric Direct Access Library Facil

ABSTRACT

GANDALF (General Alpha Numeric Direct Access Library Facility) is an information retrieval system designed and implemented at the University of New Mexico for the purposes of retrieving abstracts from large abstract data bases, such as the ERIC system. Previous batch-process information retrieval systems for use with the ERIC data base have been extremely slow, and thus expensive of computer time. Gandalf uses the user request to produce a list of addresses within the overall data base, so that only a small subset of the material is selected, and processing of unreferenced material is avoided. Furthermore, since GANDALF was designed to be used by persons with little or no computer experience, an attempt has been made to make the request statements as simple to use as possible. In comparisons runs, GANDALF was from ten to forty times as fast as QUERY (the currently available ERIC search system) in real time, and four to 77 times as fast in computer time. (Author/RH)

GANDALF

A GENERAL ALPHA-NUMERIC

DIRECT ACCESS LIBRARY FACILITY

Written at the University of New Mexico by:

Jonathan D. Embry - Southwest Research Associates

Stephen S. Baca - Department of Electrical Engineering and Computer Science, University of New Mexico

Robert Langley - Department of Mathematics, University of New Mexico

Stone Adams - Department of Mathematics, University of New Mexico

Presented to

The Rio Grande Chapter of the Association for Computing Machinery

October 13, 1972

PREFACE

The following is a portion of an in-depth report describing GANDALF. Those persons desiring more information are invited to contact the authors or:

Jonathan D. Embry
Director, Computer Services
SouthWest Research Associates
212 Bryn Mawr NE
Albuquerque, New Mexico 87106

Our thanks to Professor Don Morrison, Wilt Byrum, Ken Friedenback and the staff of the U.N.M. Computing Center and the College of Education.

INTRODUCTION

GANDALF (General AlphaNumeric Direct Access Library Facility) is an information retrieval system designed and implemented at the University of New Mexico (UNM) for the purpose of retrieving abstracts from large abstract data bases, such as the ERIC (Education Resource Information Clearinghouse) system.

The initial interest in working with the retrieval of selected information from large scale data bases the size of the ERIC system began after observing the large quantity of computer time being used by a retrieval program, QUERY, provided by the U.S. Office of Education in conjunction with the ERIC data base of approximately 75,000 abstracts on an IBM 360/67.

Upon examination of QUERY, it was noted that although the abstracts were stored on direct access devices (four IBM 2314 disk packs) the search process was sequential with no use of any type of indexes. Thus, the basic problem was determined: the data base had outgrown its access method.

It was decided that if an access method like GANDALF was to be developed, it would be far more effective if it included "modularity" as a principal feature. This modularity feature would give GANDALF the ability to access

a wide variety of data bases (including any set of machine readable records, generally in narrative or natural-language format, such as ERIC files and CHEMISTRY ABSTRACTS) with a minimal amount of modification. Sections of this paper discuss those modules and how they interface. Section describes the current status of implementation and future additions being considered to improve and extend GANDALF.

The reason for having computers process these kinds of data bases is the need to select a comparatively small group of records from a very large data base. GANDALF was designed to be used by people with little or no prior computer experience, the only user requirements being that the user have knowledge of his needs and the contents of the data base being used. The selection criterion, which is called a REQUEST, is written with one or more KEYWORDS that the desired elements of the data base contain or log cally relate to. For example, a REQUEST for the author J. Smith is actually a request for that set of records in the data base that contain the character string 'J.SMITH' in the author field. A KEYWORD, then, can be any character string that could be used to reference a record, such as COMPUTER ASSISTED INSTRUCTION, ENGLISH(SECOND LANGUAGE), and so on.

In order to reduce the volume of this paper, frequent

references are made to techniques and processes (such as reverse Polish notation and recursive programming) with which the reader may not be familiar. If this situation arises, the reader is referred to the literature as a source of background information. The authors are available for any type of additional assistance which may be desired.

OVERVIEW OF GANDALF DESIGN

The goals of the GANDALF project were to produce an information retrieval system with the following characteristics: 1) a user-oriented request language which would assist users in retrieving information and minimize user inconvenience and frustration; and 2) a system which would take advantage of third-generation equipment, especially direct access techniques, to process requests as efficiently as possible.

The user orientation was achieved by designing a new retrieval language that will be described later. The general philosophy was to eliminate as many artificial constraints as possible in input format and at the same time to allow the production of complex terms and expressions. The second objective was to reduce the large amounts of computer time spent processing extensive narrative format data bases such as the ERIC files. (See Figure 1) The second objective was achieved by designing GANDALF to build a number of indexes that are used in conjunction with a user REQUEST (a selection criterion written with one or more KEYWORDS which the desired elements of the data base contain or to which they logically relate) to produce a list of addresses that point to records in the main data base that relate to that REQUEST. These addresses are then used

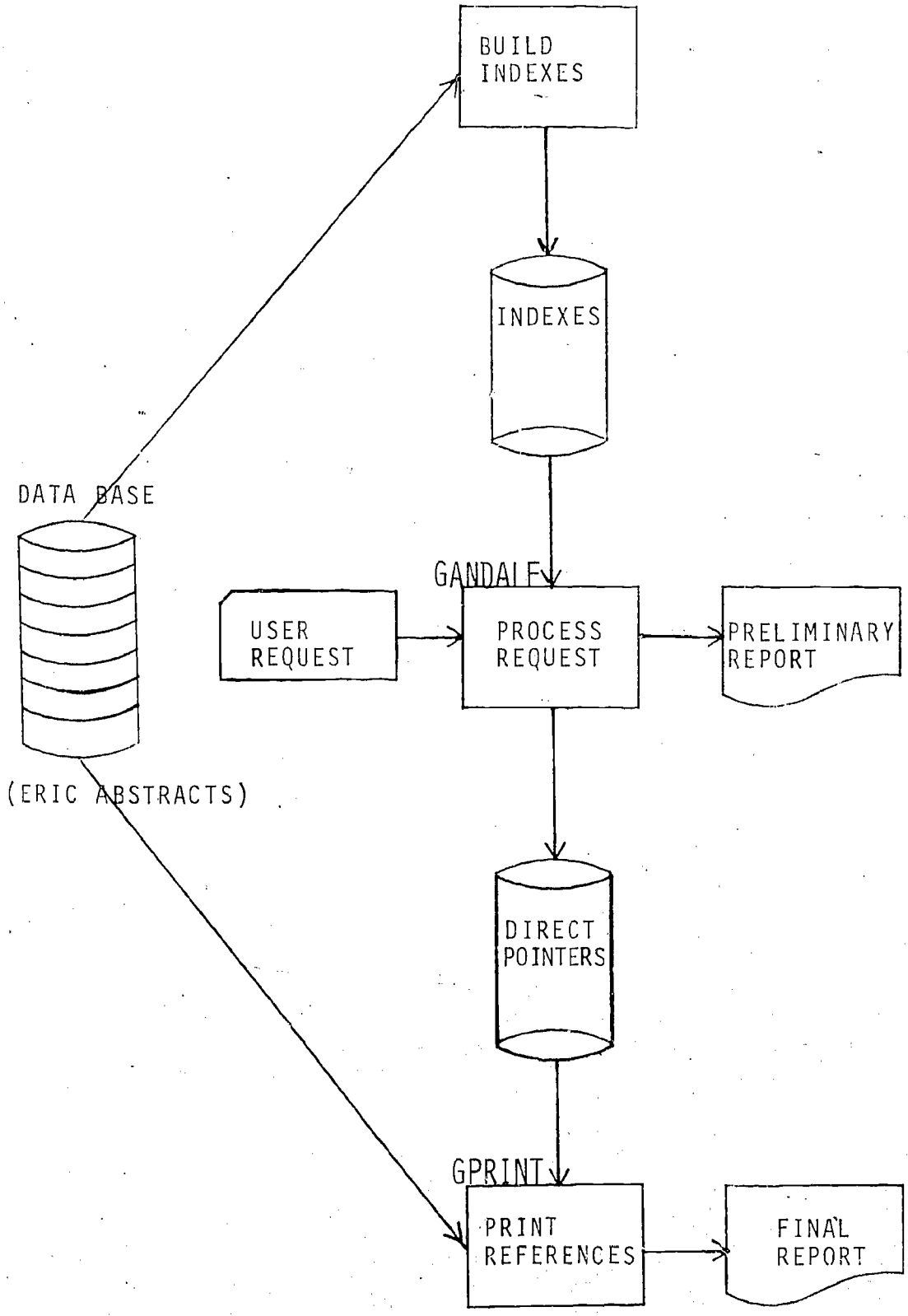
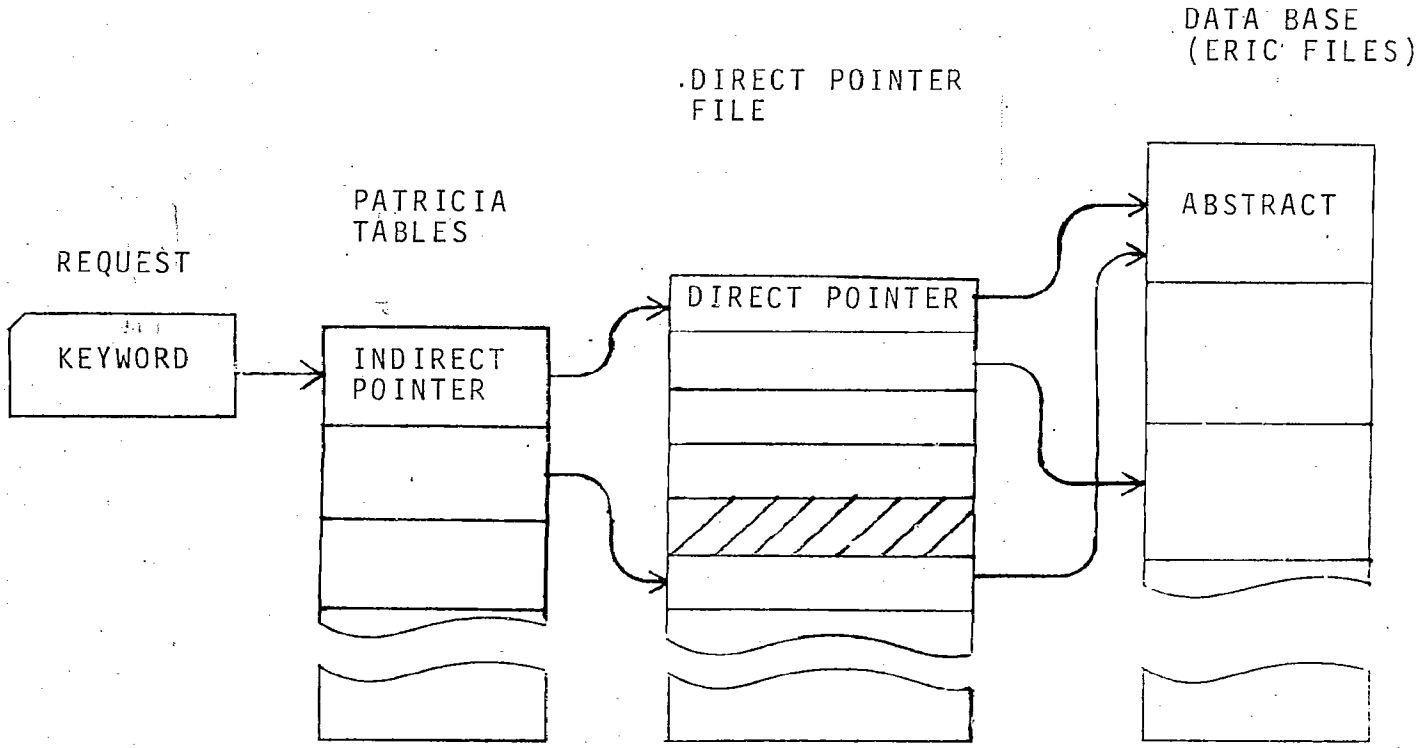


Figure 1. GANDALF overall design

to directly retrieve only those records satisfying that REQUEST without having to process unreferenced material in the data base.

The indexes built can be thought of as series of index levels (see Figure 2) where an element in one level points to a related group of elements in the next level. A KEYWORD at the first level is processed by a version of Don Morrison's PATRICIA algorithm to yield a unique number, called a PATRICIA NUMBER. The PATRICIA NUMBER points into an INDIRECT POINTER TABLE which has two elements for each PATRICIA NUMBER, one to be used if the KEYWORD is considered a primary term, that is, if it is expected to be the primary subject, author, etc., the other to be used if the KEYWORD is a secondary term. The PATRICIA NUMBER is also used as a pointer into an occurences table which has a primary and secondary entry for each member; this table describes how many times each KEYWORD occurs. Each element of the INDIRECT POINTER TABLE points to a list of elements in the DIRECT POINTER FILE. The items of a particular list point directly to records in the main data base which contain that KEYWORD. Then, any complex Boolean relation specified for a set of KEYWORDS may be evaluted by performing the appropriate Boolean operations on the lists which are associated with the different KEYWORDS. The lists are built so that the elements are in strict ascending sequence, with a special

Figure 2. Index Structure



TYPICAL SIZE OF ELEMENT	16 char/key	22 bytes/key	4 bytes/reference	2,000 char/abstract
TYPICAL NUMBER OF ELEMENTS	5 keys/request	50,000 unique keys	800,000 references	80,000 abstracts

code indicating the end of a list. The Boolean operations AND,OR, and BUT NOT essentially consist of merging two or more of these lists together. The resulting list can be used to directly access the records that satisfy that Boolean expression.

The index building process (see Figure 3) is performed once initially and then repeated every time the data base is updated (quarterly for ERIC). The KEYSEP program builds a key-reference file for each occurrence of each keyword. Each record contains a keyword as well as a DIRECT POINTER that points to the abstract that contains the KEYWORD. After being sorted alphabetically by KEYWORD, this file is used by PNTBLD. PNTBLD uses the new key-reference file merged with the key-reference file from the previous update to build the permanent DIRECT POINTER FILE and a temporary key-frequency file. The DIRECT POINTER FILE is a direct access file containing lists of DIRECT POINTERS. The key-frequency file has one record for each unique KEYWORD. Each record contains the KEYWORD, and for the primary and secondary levels, contains the number of occurrences of that KEYWORD and INDIRECT POINTERS that point to the corresponding lists in the DIRECT POINTER FILE. After the key-frequency file is sorted by frequency, it is used by PATBLD to build the PATRICIA tables. Since the records going into PATBLD have unique KEYWORDS, the PATRICIA table can be broken up into several independent segments. This allows the size of a

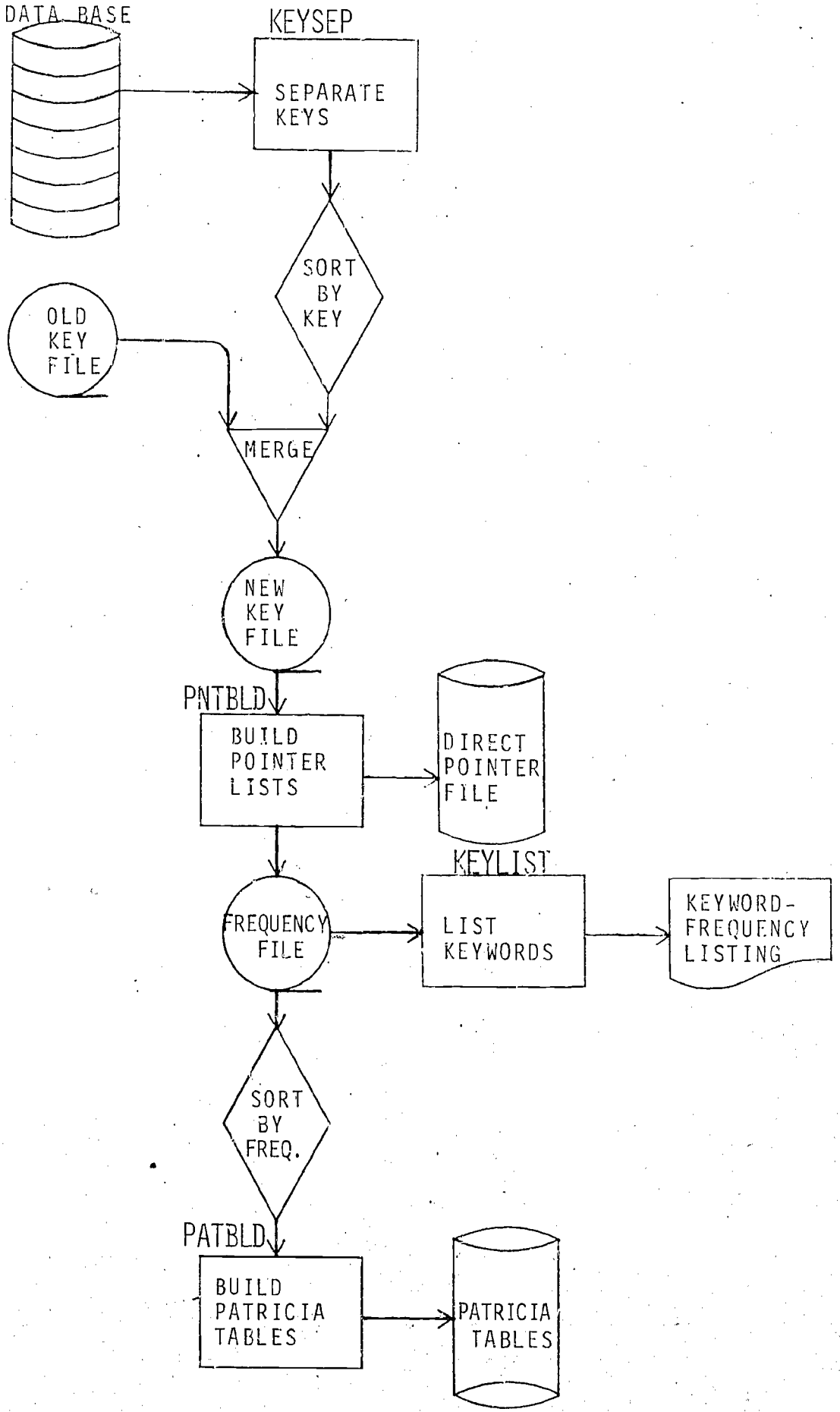


Figure 3. Index Building Process

segment to be a function of the amount of core available, since only one segment need be in [illegible] at any one time. Sorting by frequency of occur[illegible]ees that the first segment w[illegible]ll contain the most [illegible]tly occurring KEYWORDS. The underlying assumption is that most REQUESTS will be for frequently occurring KEYWORDS and can be satisfied by looking on the first few segments instead of the entire table. A further improvement on this idea is to include frequency of usage as well as occurrence as the sorting criterion. Even if the above assumption is invalidated completely, results would probably be no worse than if the table was built randomly. Thus, to the extent that requests are consistent with previous requests and the data base, the time required to look up a request will be reduced accordingly.

The retrieval program (see Figure 4) compiles a number of requests into several tables. Presently, requests are submitted in a batch mode on cards, with the future possibility of using interactive terminals or remote equipment remaining open.

Each unique KEYWORD occurring in a series of requests is added to a temporary VOCABULARY created for each run. For each KEYWORD in the VOCABULARY, an entry is made in a DIRECTORY indicating position, type and length of the KEYWORD. Simultaneously, a POLISH STRING is created, specifying the Boolean operations to be performed. For each KEYWORD in the VOCABULARY, a PATRICIA NUMBER is found using the previously built PATRICIA TABLES. The PATRICIA NUMBER is used as an index into the INDIRECT POINTER TABLE which yields a pointer to a list in the DIRECT POINTER FILE and into the occurrence table which gives the length of that list. The Boolean operations specified in the POLISH STRING are then performed on the lists that correspond to the original KEYWORDS in the REQUEST. The resultant list of record addresses is then entered into a queue of records to be printed. Whenever the main data base can be made available to the computer, the actual references can be printed (see Figure 5). The only parts of the system that are involved with the actual format or storage method of the main data base are the key separating program (KEYSEP) and the final print program (GPRINT), so that differently formatted data bases such as NASA abstracts, CHEM abstracts, and MARC records from the

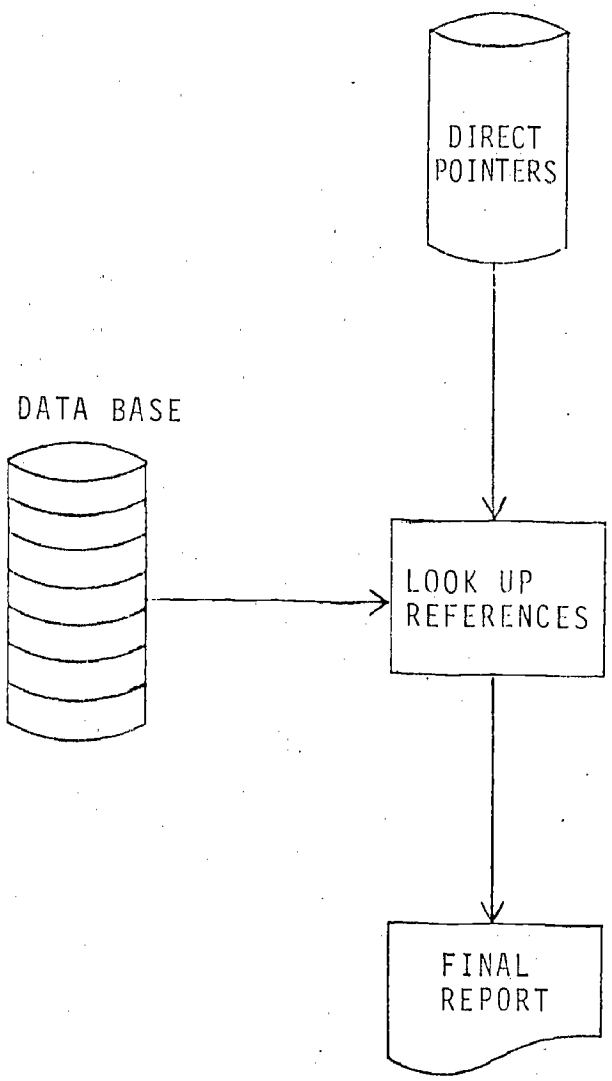


Figure 5. Produce Final Report

Library of Congress could all be processed by GANDALF in their original format by modifying those two relatively simple pieces of code; thus enhancing the flexibility of the system.

SOUTHWEST RESEARCH ASSOCIATES
P.O. Box 4092
Albuquerque, New Mexico 87106

TO: Dr. Don Morrison
FROM: Jock Embry
SUBJECT: Timing tests on GANDALF September 20, 1972

Last Saturday I ran three ERIC Searches in order to compare the different times required by QUERY (the current production program) and GANDALF. Following is a summary of the results. For each search, four times are shown. The first is the time for the initial run. For QUERY this is the time to execute a simple program that breaks the request into different jobs for each disk. For GANDALF it is the actual execution of GANDALF; that is, all processing necessary to produce preliminary reports and generate disk addresses of requested abstracts. The other three lines for each search correspond to the time required to process each disk. For QUERY, that is the time to actually process and complete a search. For GANDALF it is the time to simply retrieve and print the requested abstracts. Since virtually no work has been done on tuning and improving the GANDALF print program, those times can probably be improved considerably. Times are reported as wall clock time in minutes (CPU time in seconds).

	QUERY		GANDALF		
Search 1	.28	(.39)	.99	(3.57)	
disk 1	15.40	(70.38)	.90	(12.78)	
disk 2	14.15	(55.19)*	1.23	(18.32)	
disk 3	13.88	(60.62)	.96	(10.96)	
Total	43.69	(186.58)	4.08	(45.63)	
Search 2	.26	(.40)	1.05	(5.08)	exceeded 100 hits
disk 1	15.98	(134.86)**	0	(0)	
disk 2	16.74	(142.78)**	0	(0)	
disk 3	14.73	(117.57)**	0	(0)	
Total	47.71	(395.61)	1.05	(5.08)	
Search 3	.28	(.42)	1.12	(12.73)	
disk 1	13.83	(279.95)	0	(0)	
disk 2	13.81	(205.06)*	0	(0)	
disk 3	10.10	(240.99)	.93	(4.63)	
Total	38.02	(726.42)	2.05	(17.36)	

* abended due to disk I/O errors
** cancelled because of excessive output

From these samples it appears GANDALF is ten to forty times as fast as QUERY in wall clock time and four to seventy-seven times as fast in CPU time. Since the final results were the same in each case, I think we have fulfilled our goal of producing a more efficient retrieval system for the ERIC files.

cc: Dr. Bell
Mick McMahan
Steve Baca