

(Presented at Journées Internationales d'Informatique Médicale de Toulouse, France, March 4-6, 1970 and at The Eighth Annual Symposium on Biomathematics and Computer Science in the Life Sciences, Houston, Texas, March 23 and 24, 1970.)

## INTRODUCTION

Requirements for the design and implementation of a computer filing system in a medical research environment are nearly the same as those encountered in general data processing. There are data sets of many varieties and sizes, some data have to be kept over long periods, and some require regular updating. Loss of data, whether caused by a system failure or by a human error, can be costly and frustrating.

These conditions become more severe when a flexible filing capability is being offered to medical researchers who are doing their own programming and operating solely in an on-line mode. Normally, the professional programmer is responsible for insuring data integrity, and he achieves that by careful writing of procedures and by generating frequent backup files. In an on-line system, it is important that the same amount of careful programming is not required and that backup is provided automatically. Furthermore, medical research personnel are often inexperienced in computer techniques and do not wish to be concerned with the intricacies of operating systems and the hardware when storing or retrieving information.

This paper describes a filing system as designed and implemented by the ACME (Advanced Computer for MEDical Research) Facility. User language statements, the internal structure of the data organization, and methods by which reliability is achieved are summarized. Charts are presented which illustrate usage patterns of the system after more than two years of operation. These charts are seen to be especially useful to designers of new systems in which similar support facilities are provided. Since the file system has relatively liberal restrictions as to record size, total file size, data types, and access methods in comparison with other on-line systems now operating, we hope that extrapolations to other environments can be made.

## ACME CONFIGURATION

ACME is a typewriter terminal driven, time-sharing system designed and implemented by the Real-Time Facility of the Stanford University Computation Center for the Stanford University School of Medicine. The purpose of ACME is the acquisition, analysis, storage, retrieval of medical research data,

and the control of laboratory instruments. Emphasis is placed on real-time data acquisition and on on-line control of instruments [1]. The system provides a simple, yet relatively powerful subset of the PL/1 language called PL/ACME [2].

The current hardware configuration is shown in Figure 1. It is capable of supporting 30 users and 24 data acquisition lines at any time. An IBM 360, Model 50, with 2176K bytes of core memory, is responsive to a variety of input and output devices. User access is via a 2741 typewriter terminal. User programs and data are stored on two 2314 multi-disk units. Laboratory instruments may be interfaced through an IBM 1800 analog/digital computer, a 2701 high-speed, data acquisition controller, or a port in an especially designed controller known as the 270X. Data transmission links to the Stanford Computation Center are also available.

Figure 2 is a simpler schematic of ACME's hardware with emphasis on the flow of data to and from the central processor. The nucleus of IBM's Operating System and the many ACME routines are core resident while the system is operational. The remainder of available core (1500K bytes) is allocated to user programs and data. Only the 2314 disk units are time-shared for data storage. Magnetic tapes serve as backup to the disk packs and as archival storage for user data which need not be immediately accessible.

## DEFINITIONS

Figure 3 summarizes the definitions of several terms used throughout the remainder of this paper.

A RECORD is the basic unit for storage of information. It may contain as data a single variable, an array, a structure, or a text line. It is identified by a record number or KEY. This may be an integer or fractional number with up to three decimal places.

A DATA SET is a collection of records identified by a user assigned name. The system refers to a data set by appending two more names to the data set name: the user name and his project name. This produces a qualified name of the form: 'User name.Project name.Data set name'. In addition, a unique data set number is assigned. Thus, a qualified data set name or data set number and a record number uniquely identify a record. Figure 4 shows the storage hierarchy of a data set.

When a data set is opened by a program, the system builds a control area in core memory. The set of information consisting of the data set, its control records (index, directory), and the control area in core constitutes a FILE.

The storage volumes are divided into fixed-length units of space called BLOCKS. Each block is assigned to one and only one data set and may contain one or more records, a catalog of names, etc.

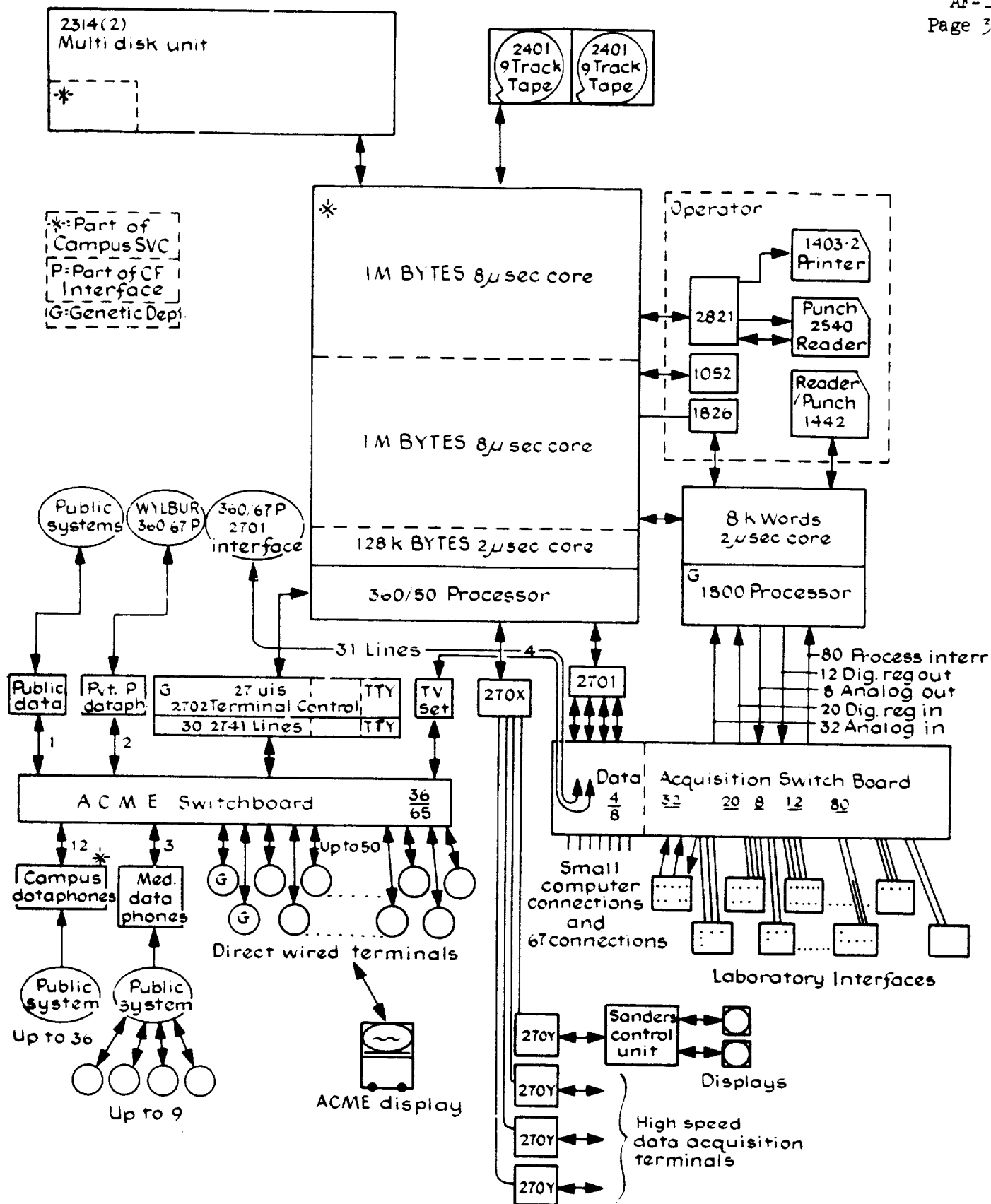


Fig.1 ACME Hardware Configuration

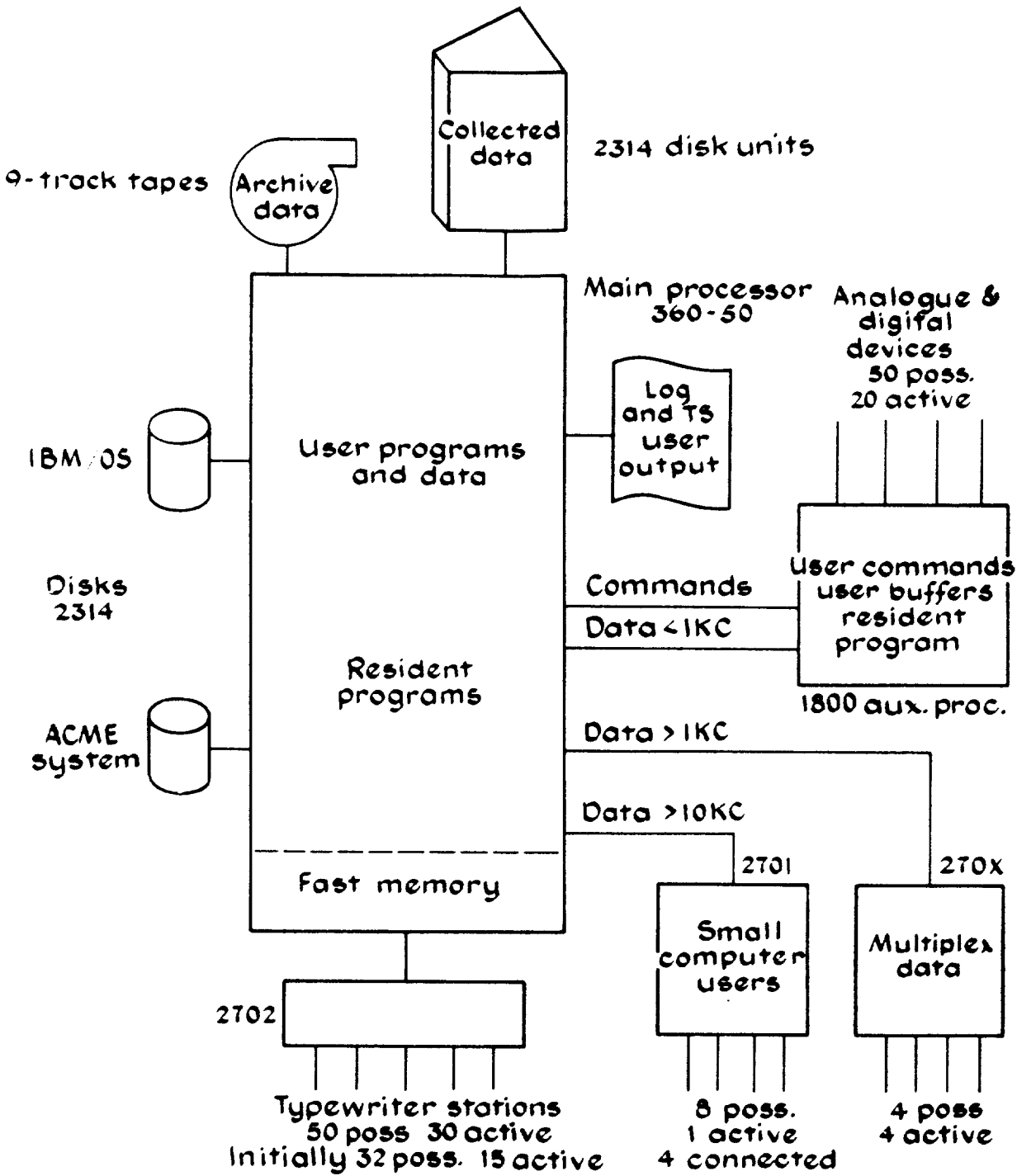


Fig. 2 Data flow in ACME

The space requirement for data and text files represents 65.4% of the total storage with an aggregate waste of 26.9%. This efficiency of use compares favorably with other time-shared or manufacture-supported file systems.

## APPLICATIONS

There are approximately 200 active projects at ACME. The majority involve studies in medical research or other medically related disciplines. Only a dozen projects have no relevance to medicine. Most projects are located in the various departments of the Stanford Medical Center. Ongoing projects also exist at the Palo Alto Medical Research Foundation, the Palo Alto Veterans Administration Hospital, and the University of California Medical Center in San Francisco.

User projects vary from problems in basic medical research to daily clinical diagnosis and administration(see Figure 20). We have a number of research projects in which data is collected in a real-time mode, analyzed, and filed or displayed on various output devices. Real-time data acquisition is an important part of such projects as Cardiology's artificial heart study, Anesthesia's project on respiratory control systems, and Pathology's auto-analysis of blood samples [7].

Clinical applications usually combine patient diagnosis with research in some area of medicine. An example is Dr. Petralli's MED DATA project in Infectious Diseases [8]. This project has a dual purpose: To improve the quality of laboratory antibiotic sensitivity reports and to collect information on antibiotic sensitivity patterns which may be of value in treating rarely encountered organisms. Two daily files are maintained. The first contains technician reports on today's culture samples. A quality control program processes these reports and produces a printed daily summary and a second data set in which all errors have hopefully been eliminated. The edited data set is then merged with an accumulative data set which has been sorted by organism. From the sorted data set, histograms are plotted showing the frequency of occurrence of various zone sizes in the treatment of an organism with a specific antibiotic. These histograms are used to improve the quality control program and to guide the physician in his selection of antibiotics.

Some examples of other data on file include blood typing information for kidney transplant recipients, the effect of radiation therapy on terminal cancer patients, census information for studies in genetic drift, and 40,000 points taken from a mass spectrometer analysis of 20 grams of moon dust. From the latter it can only be inferred that the moon has no organic components.

The largest and most complex data set is used by the Clinics Business Office for fast access to patient billing information [9]. This data set consists of 13.2 million characters of information on 36,000 clinic patients. An inquiry program is loaded into the CPU memory each morning and upon demand displays on a Sanders 720 graphic display any of 67 different items of information on a patient.

**MEDICAL RESEARCH**

Real-time data acquisition  
Laboratory instrumentation

**CLINICAL AND LABORATORY DIAGNOSIS**

Patient services  
data analysis and interpretation

**EDUCATION**

Student research projects  
Student self-teaching

**MEDICAL ADMINISTRATION**

Data quality and quantity control  
Patient information retrieval

**APPLICATIONS**

The data set is organized by a 6-digit patient hospital number. Minimum access time for any patient is about one second; mean time is five seconds.

In reviewing the various applications of ACME and its file system, we discovered that most users simply do not write complex data sets. The reasons are several: Many projects do not require elaborate filing schemes; few users have had extensive experience in computer programming and thus do not know how to organize a complex file; and ACME does not as yet provide generalized indexing and sorting routines for ordering data.

Everyone likes text files. They are easy to search, sort, and update. A population study project in Pediatrics writes field reports in text form which we then list on the line printer in 'clean' mode, i.e., with the line numbers omitted.

Record size is currently limited to the number of characters which can be contained in one block. This restriction is being removed so that one WRITE statement could, if so desired, write all of our core memory as one record on the storage device.

PL/1 structures were implemented a few months ago and enthusiastically welcomed. Previously, a record could be written from one variable. With structures, a record may be written as a collection of variables. An example of an application is a laboratory report where test results are stored numerically but must be correlated with the patient identification which is in character form.

## CONCLUSION

In conclusion, we can only say that, while all is not perfect, we do feel that we have a filing system that provides most of the features which are required by the ongoing projects at the Stanford Medical Center. The exception is cheap storage. The 2314 is costly. However, incorporation of a data cell, while providing storage at  $\frac{1}{3}$  the cost of an equivalent block on a 2314, would result in an access time roughly five times slower. And in a real-time environment, fast storage and retrieval are crucial. Perhaps our hardware manufacturer will eventually develop a storage device that is fast, easily rewritten, and inexpensive. Additionally, we are aware of requirements for medical records that do not include the rewrite capability.

The ACME filing routines were written with several criteria in mind: ease of use, speed of operation, flexibility, maintainability, and reliability.

Ease of Use. The PL/ACME statements for reading, writing, and updating records could not be simpler. The user need not be concerned with defining to the Operating System the characteristics of his data set such as the access volume, the space requirements, or the disposition. If the data set does not exist, we create it. If it does exist, we will add records to the present data set.

Speed of Operation. Our access times compare favorably with those provided by other time-sharing systems. While the time required for an I/O operation depends heavily upon the system load, a write time of 100 milliseconds with only a few active users can be expected.

Flexibility. Only the imagination of the user restricts his potential application of available storage. The PL/ACME language contains many string and array manipulation functions which allow him to order his data in a variety of sequences.

Maintainability. The user has few problems when he updates a record. If the new record is of the same or lesser length as the previous, the same space will be reused; if not, we will find a new block for it.

The ACME programmer staff can depend upon the various dumping, analyzing, and restoring routines which readily indicate system status and provide a convenient method for repairing errors.

Reliability. Perhaps this is our strongest argument for the present design. In spite of several hardware and operations failures in the past two years, we know of only two instances in which data was irretrievable. In the first instance, modifications made on that day to an existing program were lost. In the second, a complete program was destroyed, but another copy existed under a different name and project. No patient data, once it had been filed, has ever been lost. Most of the overhead involved in the file system is concerned with repeated software redundancy checks. Is the block truly available? Does the block really contain records belonging to this data set? Many computer hours are spent on backup dumping and analysis. Costly, but it's worth it.

This work was supported by the Josiah Macy Foundation and the National Institutes of Health (Grant FRO311). The overall design of the ACME File System is due to Jerry Miller[4].



## REFERENCES

Internal ACME documentation is available upon request. An ACME Notes Index, AA, lists all ACME notes. To obtain copies of any note, write to the authors at the ACME Facility, Stanford Medical Center, Stanford University, Stanford, California.

1. Wiederhold, G., "An Advanced Computer for Medical Research," published in the Proceedings of the IBM Japan Computer Science Symposium--Research and Development and Computer Systems, Tokyo, Japan, November 1969, pp. B1-B15.
2. Breitbard, Gary Y. and Gio Wiederhold, "PL/ACME: An Incremental Compiler for a Subset of PL/1," IFIPS 68 Conference, Edinburg, Scotland, August 1968.
3. Wiederhold, Voy, "How to Use PL/ACME," ACME Note AM, September 25, 1968.
4. Miller, Jerry, "The ACME File System," ACME Note FY, February 27, 1969.
5. Girardi, Serge, "ACME File Input/Output," ACME Note FIO, May 8, 1969.
6. International Business Machines Corporation, "IBM System/360 Operating System, System Programmer's Guide," SRL C28-6550.
7. Crouse, Linda P. and Gio Wiederhold, "An Advanced Computer for Real-Time Medical Applications," Computers and Biomedical Research, Vol 2, no. 6, December 1969.
8. Petralli, J. K., S. Wallis and T. C. Merigan, "A Computer Method for Improvement of Antibiotic Sensitivity Data and Guidance in Therapy," Clinical Research, January 1969.
9. Frey, Regina, "Clinics Business Office Inquiry System," ACME Note CBO, 1970.