Dartmouth College HANOVER · NEW HAMPSHIRE · 03755

Kiewit Computation Center

DTSS -- THE DARTMOUTH TIME-SHARING SYSTEM

Introduction

The Dartmouth Time-Sharing System is a coordinated group of programs which form a complete software system for the Honeywell 600/6000 series computers. The objective of the system is to provide general purpose computing facilities for users having remote access to the computer through teletypes or other functionally similar terminals. The system is designed to handle over 150 users simultaneously and to have a response-time lag of less than five seconds for moderate-size edit-compile-and-go problems. This system is developed to satisfy the expanding requirements of the College for computing in administrative work, as an aid to education, as a tool for research in many fields, and as a medium for research in computer science itself.

The software development project is directed by John S. McGeachie, together with two full-time system programmers. Three additional Dartmouth faculty members devote time to the project. One or two men from outside the College have usually been involved on the project, normally for periods of six months or less. Two graduate students worked on the system for approximately one year. The bulk of the programming effort has been undertaken by Dartmouth undergraduates under the supervision of faculty members involved in the software development effort. These students have worked part-time at the computing center during the academic year and full-time during the summer recess. This programming activity has been entirely extra-curricular; the students have carried a normal undergraduate course load at all times.

Design

The design of the system is based in large part on the successful experiences Dartmouth has had with the GE-235/Datanet-30 coupled computer system developed at the College in 1963 and in operation until 1967. The activities of users on this system were sampled and the new DTSS has been designed to give good response for the most frequent forms of user activity. On the other hand, it was realized that the 235/D-30 system was not general enough to allow easy growth of facilities such as compilers or file systems, so a demand for a more general time-sharing system developed.

It was also recognized that a balance is required between a simple, fast-response, limited utility service, and a system which provides the most general sort of programming facilities. These general capabilities are seldom used and can lead to inefficiencies which are in conflict with the responsibility of the system to provide good service to a large class of users who are using the system for simple edit-compile-and-go problems from remote teletype terminals. There is also a conflict of interest from the user's point of view if he is required to provide exhaustive lists of parameters to a general purpose time-sharing system to perform such a simple operation as edit-compile-and-go. In such a case, the user must understand the general capabilities available just so that he may bypass them or use them trivially. This conflict was resolved in the design of DTSS.

DTSS is based on a very general executive system which provides a simple basis for the easy growth of sophisticated computing services. The user is presented to a monitor -- a job running in the framework of the general executive system; the monitor specifies the parameters, files, etc. to the executive system automatically and communicates with the user. The Simple Monitor (SIMON) is a monitor which provides the simple, successful service that made the 235/D-30 system popular, and is the monitor accessed by most DTSS users.

Monitors have been developed for background and remote batch services, and a simulator has been developed under which the Honeywell GCOS batch processing system can be run. A special-purpose monitor has been developed for use by the operations staff in the day-to-day maintenance of the system. This monitor is called OPMON and actually is identical to Simple Monitor but is run in a very privileged environment. Figure 1 shows the user interface with the various monitors.

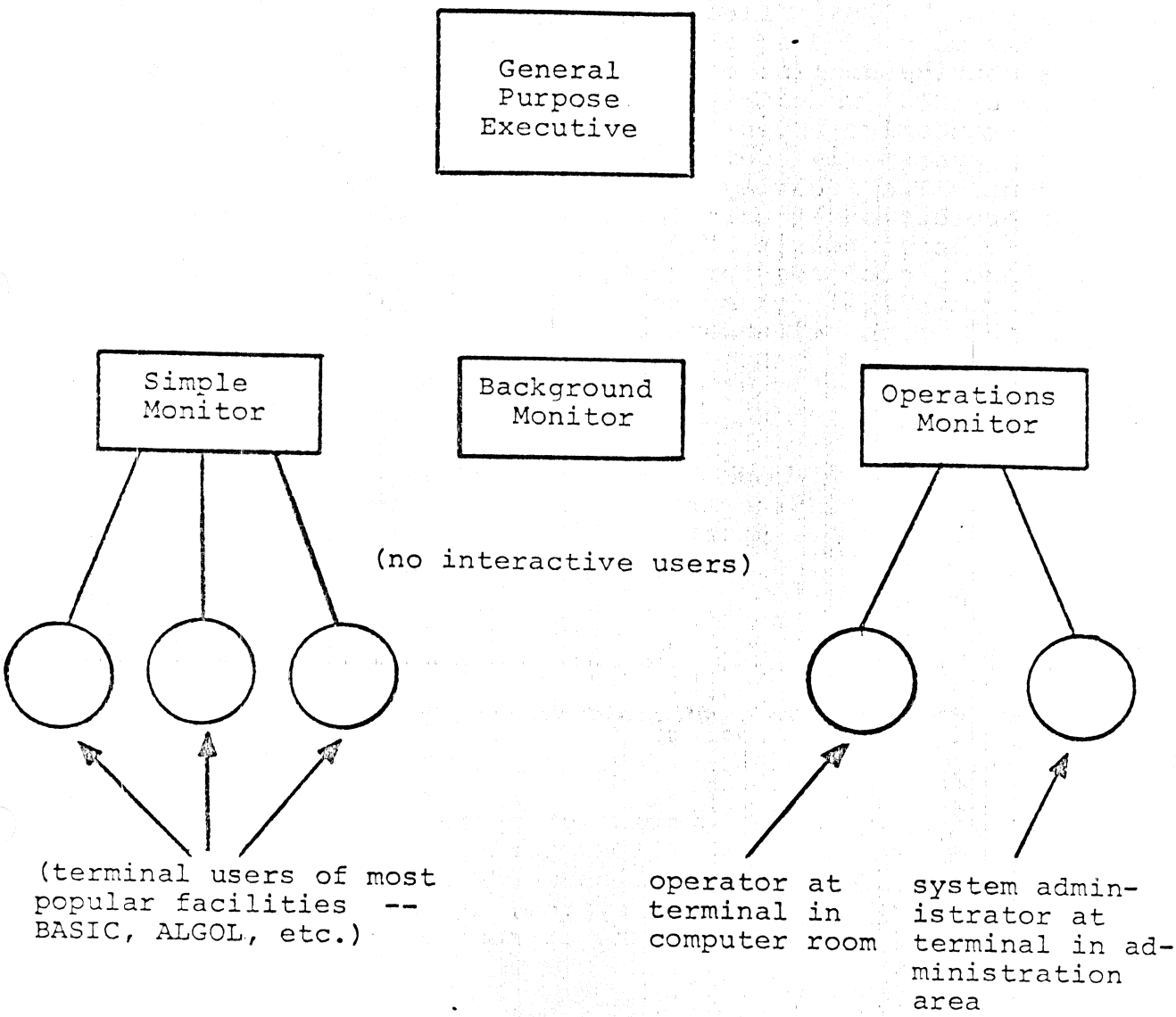

General Purpose Executive
Simple Monitor
Background Monitor
Operations Monitor
(no interactive users)
(terminal users of most popular facilities -- BASIC, ALGOL, etc.)
operator at terminal in computer room
system administrator at terminal in administration area

Figure 1

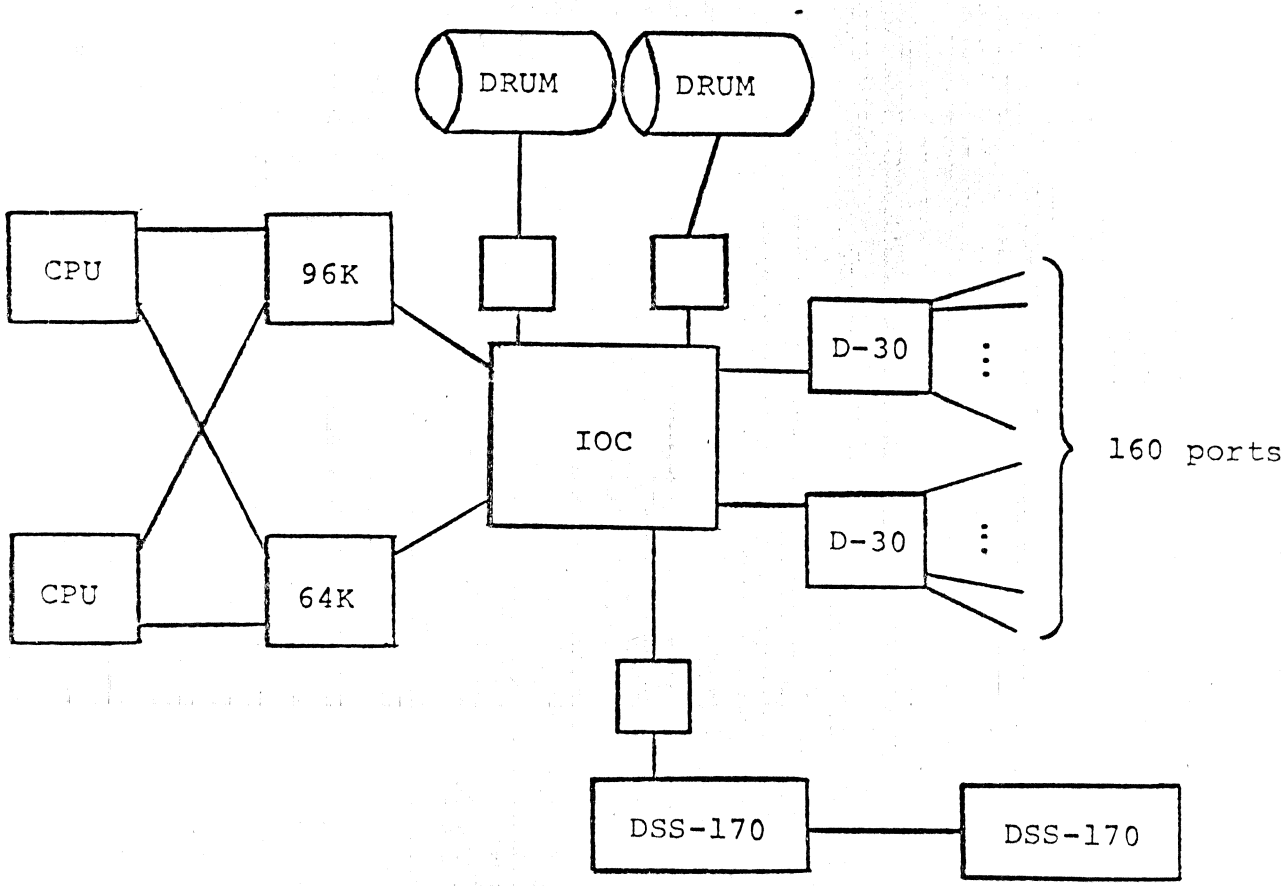
File System

The fundamental rock upon which DTSS is constructed is the file system. DTSS files and the catalog tree reside on permanently mounted disc and drum storage. A file in DTSS is simply a contiguous list of words. File storage is automatically allocated by DTSS when the file is written into; all files may be addressed randomly. Even to the assembly-language programmer, the file system is device independent and record-length independent. The security of the file system is effected by password protection at each level of the catalog tree and by trap programs. The integrity of the file system is assured by programs and procedures for dumping the logical or physical file structures onto reels of magnetic tape. The complete file system is accessible from all subsystems of DTSS.

Hardware

The hardware on which DTSS operates is a Honeywell 600/6000 series computer. DTSS operates with machines providing from 64K to 256K words of core storage, and the software is designed to support from one to eight central processing units. DTSS supports from one to four Datanet-30 communications controllers, disc systems such as the DSS170 or DSS180, and drum-like devices such as the MDU200 or DSU270. Seven- and nine-track magnetic tape units may be used in addition to the standard unit-record peripherals such as the line printer, card reader, and card punch. A large 6050 configuration for 160 simultaneous users would rent for approximately $50,000 per month.

DTSS at Dartmouth currently operates on a dual-processor Honeywell 635 system with 160K words of core. Two DSS-170 storage devices provide 72 million words of disc storage. There are 160 ports available on the system, at speeds ranging from 110 to 2000 bps. Figure 2 shows the current hardware configuration.



Dartmouth Time-Sharing System
Honeywell 635 Hardware Configuration

Figure 2

DTSS--CAPACITY

Daily Throughput

DTSS processes more jobs in a single day than most computer centers process in an entire month. As many as 20,000 jobs have been processed by DTSS in a single 17-hour period. Note that 1% of these are long running jobs requiring several minutes of processor time. At Dartmouth 1000 different users log in for 3000 sessions each day. In an active month 5000 different persons use DTSS.

Wide User Base

There are 25,000 valid user numbers on the version of DTSS at Dartmouth. These users include 50 colleges and high schools distributed throughout seven eastern states and the Province of Quebec in Canada. DTSS also services a small number of commercial firms in the northern New Hampshire and Vermont areas. The system provides useful services to a wide spectrum of users. More than 90% of the 3000 undergraduates at Dartmouth use DTSS in the academic year.

Simultaneous Users

DTSS supports more simultaneous users than any other time-sharing system. Averaged over all the hours of system operation, the average number of simultaneous users is 50. Hardly a day goes by in which the peak number of simultaneous users is not greater than 100. The system has in fact on occasion handled as many as 137 users with response times to most commands remaining under ten seconds. The hardware installed at Dartmouth will allow up to 160 users if the demand requires this. DTSS software will allow about 250 users.

Resource Usage

On a typical day 1000 hours are logged by users of DTSS at terminals. Eight hours of processor time are used and billed. Nearly a half-million accesses to the disc storage systems are completed in the same period.

Cost Effectiveness

The cost effectiveness of DTSS is due largely to the capacity of the operating system to allocate the minimum number

of resources necessary to perform a given task. The software system was designed to provide service to 200 users; at each step of the design the question was asked -- "What would be the effect of 200 users on the system?"

DTSS--AVAILABLE LANGUAGES

BASIC

BASIC was developed at Dartmouth in 1964. Although it was originally designed with emphasis on ease of learning, it has continually been expanded to become DTSS BASIC -- a language including such features as

- data packing and unpacking
- direct access file support
- independent subprograms
- string handling
- interactive symbolic program debugging.

FORTRAN IV

FORTRAN IV on DTSS is second only to BASIC in popularity. DTSS FORTRAN IV contains several additional features which allow file access in a manner compatible with BASIC. A new FORTRAN language processing system is under development; it contains the full ANSI standard as a subset.

COBOL

A COBOL system has been developed which will allow rapid construction and debugging of COBOL programs from remote terminals. DTSS COBOL is full ANSI COBOL 68 with the exception of the segmenting feature. The COBOL compiler and run-time package have been verified using the Navy-prepared audit programs.

ALGOL 60

ALGOL 60 is also available on DTSS. The version available on DTSS follows "The Revised Report on the Algorithmic Language ALGOL 60" (Communications of the ACM, January 1963) with some commonly accepted exclusions. In addition DTSS ALGOL 60 contains

- double precision capability
- string handling facilities
- input/output for terminals and files
- chaining to successive ALGOL 60 programs.

ALGOL 68

ALGOL 68 is an extensible language of the ALGOL type, having features for the manipulation of structured data, strings of characters, bits, etc. Unlike PL/1, it is a safe language which insures at compile-time that an incorrect program cannot destroy data or sequencing information. ALGOL 68 is under development although some limited experimental use is possible.

DTRAC

DTRAC is the DTSS version of a published language for Text Reckoning and Compiling. It is essentially a character-handling language with a macro facility.

LISP 1.5

LISP 1.5 is an interpretive language for the manipulation of list structures. The DTSS LISP 1.5 interpreter most closely resembles that implemented at one time by the MIT Project MAC artificial intelligence group.

MIX

The MIX language system implements the ficticious computer used to illustrate computer operation by Donald Knuth in his series of books The Art of Computer Programming. A machine language interpreter, a debugger, and a symbolic assembler exist.

GMAP

GMAP is the assembly language for the Honeywell 635 which is the host computer for DTSS. This is a high-speed, two-pass macro assembler with excellent cross-reference features. Assemblers are also available for Honeywell's Datanet-30 Communications Computer (DMAP), Digital Equipment Corporation's PDP-9 (9MAP), Data General's NOVA (NMAP) and IMLAC Corporation's IMLAC (IMAP).

DDT

DDT is a system for debugging machine language programs produced by the GMAP assembler. DDT is highly developed, with selective breakpointing, symbolic patching, and file processing abilities. This technique was used to debug the entire DTSS software system.

Editors

A variety of editing systems are available to users of DTSS. One is oriented to process files of ASCII characters arranged in the line number order which most users prefer. Another editor operates on context, and a third allows the flexibility of regular string expressions and programs of edit commands.

DTSS--TIME-SHARING BATCH-PROCESSING CONTINUUM

Time-Sharing

DTSS is most widely known for its time-sharing capability. Indeed the first goal of DTSS was to minimize the effort required for a user at a terminal to compose and execute computer programs. Such interactive usage has expanded through the development of sophisticated data analysis packages, file-system maintenance procedures, and hardware maintenance aids.

Background

The background mode of operation is available for jobs which require long running times or other resources not normally available to interactive users. These resources would include the magnetic tape drives, the high-speed printer, etc. The user may write and debug a program in the interactive mode and then submit a background job description to schedule the job to be executed at the discretion of the Background System Monitor. The job specification statements are checked immediately upon submission of a job, so that simple job control statement errors are immediately brought to the attention of the user. Status reports are appended to the job description file as the work progresses so that a user at a terminal may determine, for example, if he should go to the computing center to pick up his output. This illustrates how the best features of foreground and background operation are made available to the DTSS user.

GCOS

The traditional operating system for the 600/6000 line is GCOS. Users of DTSS who also desire GCOS III services may obtain them through the GCOS III Subsystem Simulator which operates under the Background Monitor. The expanded error messages presented under this mode of operation are only one of the advantages of operating GCOS under DTSS.

Users may submit decks of cards in the traditional manner, or they may indicate a file which contains the GCOS III job. This mode of operation enables all of the capabilities of DTSS to be brought to bear on the problem of constructing and updating this job submission file. An important advantage is that most control-card errors are detected at the time of submission of a GCOS job. Another benefit is that reports may be appended incrementally to a file as job steps are completed, thus allowing

a user at a terminal to follow the progress of a GCOS III job through the system. A side benefit is the reduction in card handling by the operator; DTSS operates with only one operator.

DTSS--TERMINAL SUPPORT

Terminal Support

DTSS supports all popular time-sharing terminals. Software has been developed over a period of five years for the Datanet-30 communications controller which makes economical, reliable support possible for the following terminals:

Teletype 33/35	110 bps
Teletype 37	150 bps
IBM 2741 (EBCDIC)	134.5 bps
IBM 2741 (Corrsp)	134.5 bps
Terminet-300	300 bps
Memorex 1250	600 bps
Teletype Inktronic	1200 bps
Remote Batch Terminal	2000 bps

DTSS can operate with as many as 250 terminals; it has been tested successfully with 160 ports operational.

Applications

The prolific development of system services has been exceeded only by the development of applications services by the thousands of students, faculty, administrators, and others who number among the 25,000 active users of DTSS. The development of advanced applications programs has placed demands for expanded system services upon the DTSS development staff at Dartmouth. This peculiar symbiosis between applications development and DTSS development has resulted in a balanced, cost-effective resource available to the Dartmouth community.

For example, although most files on DTSS are ASCII, special packing and unpacking facilities were made available in BASIC to enable a survey summary system to be developed using random-access files. The IMPRESS (Interdisciplinary Machine Processing for Research and Education in the Social Sciences) application has won national acclaim for the facile manner in which hypotheses of social interaction may be tested against real data.

The easy use of foreground and background processing for data base management has encouraged the development of traditional data processing applications in the flexible environment of DTSS. Project management has been implemented on DTSS, and several management information systems have been put into use with good results.

Libraries of hundreds of applications programs are currently available on DTSS, and are constantly being expanded and augmented. In addition to programs, supporting curriculum materials for undergraduate education are prepared under the auspices of several projects underway at Dartmouth.

Dartmouth College HANOVER · NEW HAMPSHIRE · 03755

Kiewit Computation Center

TECHNICAL MEMORANDUM TM022

TO: All Users of the Dartmouth Time-Sharing System

FROM: R. Brough
R. Hargraves
D. Mather

DATE: September 1972

USER'S GUIDE TO THE DARTMOUTH TIME-SHARING SYSTEM

SUBJECT:

ABSTRACT: This manual is intended to help users of DTSS sign on to the system and interact with the computer in order to create new information and to alter information previously stored. In addition to providing instructions for signing on the DTSS, and creating and editing files, it contains a complete description of the system commands available to all users and an explanation of system error messages.

SUPERSEDES: June 1972 edition by R. Brough, R. Hargraves, and D. Mather

TABLE OF CONTENTS

Dartmouth College HANOVER · NEW HAMPSHIRE · 03755

Kiewit Computation Center

DTSS -- THE DARTMOUTH TIME-SHARING SYSTEM

Introduction

The Dartmouth Time-Sharing System is a coordinated group of programs which form a complete software system for the Honeywell 600/6000 series computers. The objective of the system is to provide general purpose computing facilities for users having remote access to the computer through teletypes or other functionally similar terminals. The system is designed to handle over 150 users simultaneously and to have a response-time lag of less than five seconds for moderate-size edit-compile-and-go problems. This system is developed to satisfy the expanding requirements of the College for computing in administrative work, as an aid to education, as a tool for research in many fields, and as a medium for research in computer science itself.

The software development project is directed by John S. McGeachie, together with two full-time system programmers. Three additional Dartmouth faculty members devote time to the project. One or two men from outside the College have usually been involved on the project, normally for periods of six months or less. Two graduate students worked on the system for approximately one year. The bulk of the programming effort has been undertaken by Dartmouth undergraduates under the supervision of faculty members involved in the software development effort. These students have worked part-time at the computing center during the academic year and full-time during the summer recess. This programming activity has been entirely extra-curricular; the students have carried a normal undergraduate course load at all times.

Design

The design of the system is based in large part on the successful experiences Dartmouth has had with the GE-235/Datanet-30 coupled computer system developed at the College in 1963 and in operation until 1967. The activities of users on this system were sampled and the new DTSS has been designed to give good response for the most frequent forms of user activity. On the other hand, it was realized that the 235/D-30 system was not general enough to allow easy growth of facilities such as compilers or file systems, so a demand for a more general time-sharing system developed.

It was also recognized that a balance is required between a simple, fast-response, limited utility service, and a system which provides the most general sort of programming facilities. These general capabilities are seldom used and can lead to inefficiencies which are in conflict with the responsibility of the system to provide good service to a large class of users who are using the system for simple edit-compile-and-go problems from remote teletype terminals. There is also a conflict of interest from the user's point of view if he is required to provide exhaustive lists of parameters to a general purpose time-sharing system to perform such a simple operation as edit-compile-and-go. In such a case, the user must understand the general capabilities available just so that he may bypass them or use them trivially. This conflict was resolved in the design of DTSS.

DTSS is based on a very general executive system which provides a simple basis for the easy growth of sophisticated computing services. The user is presented to a monitor -- a job running in the framework of the general executive system; the monitor specifies the parameters, files, etc. to the executive system automatically and communicates with the user. The Simple Monitor (SIMON) is a monitor which provides the simple, successful service that made the 235/D-30 system popular, and is the monitor accessed by most DTSS users.

Monitors have been developed for background and remote batch services, and a simulator has been developed under which the Honeywell GCOS batch processing system can be run. A special-purpose monitor has been developed for use by the operations staff in the day-to-day maintenance of the system. This monitor is called OPMON and actually is identical to Simple Monitor but is run in a very privileged environment. Figure 1 shows the user interface with the various monitors.

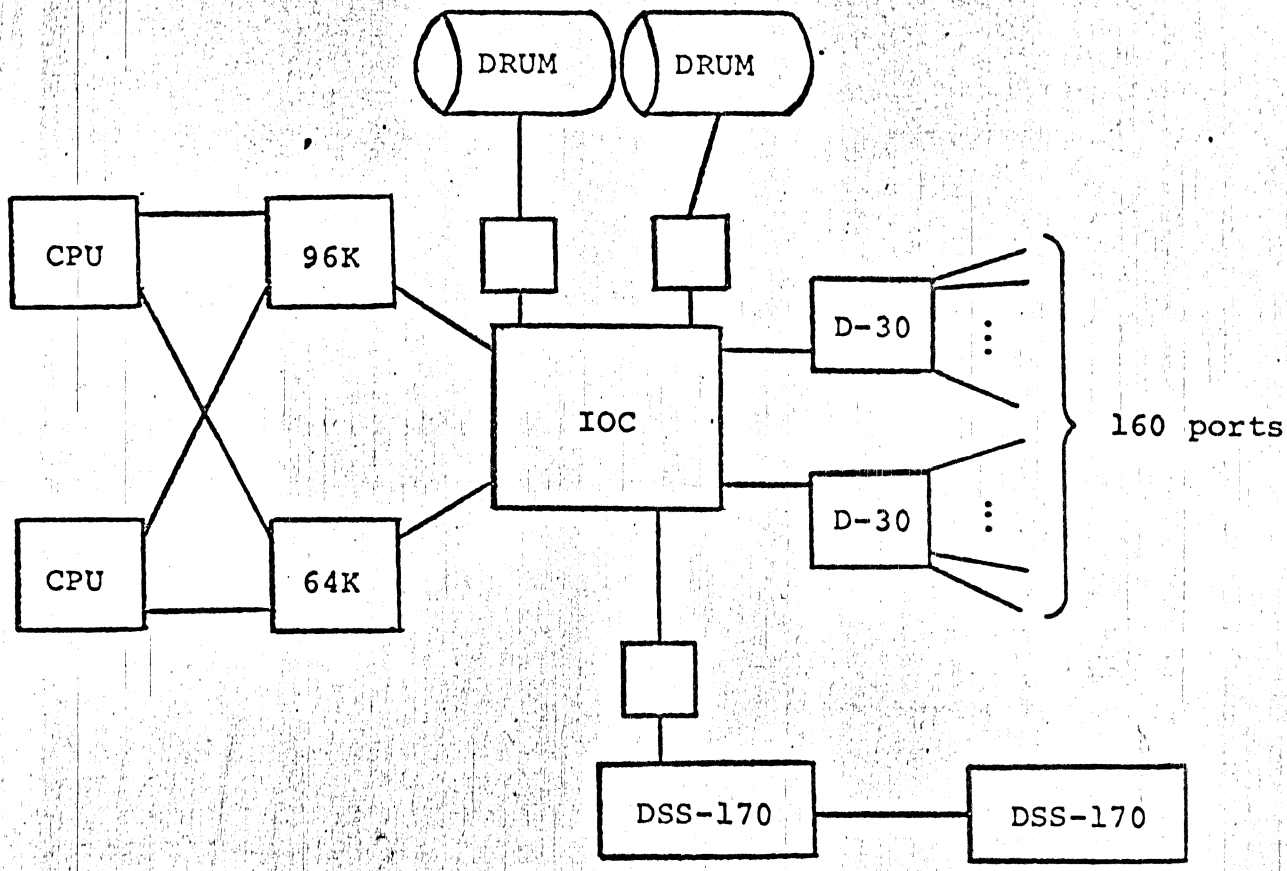
File System

The fundamental rock upon which DTSS is constructed is the file system. DTSS files and the catalog tree reside on permanently mounted disc and drum storage. A file in DTSS is simply a contiguous list of words. File storage is automatically allocated by DTSS when the file is written into; all files may be addressed randomly. Even to the assembly-language programmer, the file system is device independent and record-length independent. The security of the file system is effected by password protection at each level of the catalog tree and by trap programs. The integrity of the file system is assured by programs and procedures for dumping the logical or physical file structures onto reels of magnetic tape. The complete file system is accessible from all subsystems of DTSS.

Hardware

The hardware on which DTSS operates is a Honeywell 600/6000 series computer. DTSS operates with machines providing from 64K to 256K words of core storage, and the software is designed to support from one to eight central processing units. DTSS supports from one to four Datanet-30 communications controllers, disc systems such as the DSS170 or DSS180, and drum-like devices such as the MDU200 or DSU270. Seven- and nine-track magnetic tape units may be used in addition to the standard unit-record peripherals such as the line printer, card reader, and card punch. A large 6050 configuration for 160 simultaneous users would rent for approximately $50,000 per month.

DTSS at Dartmouth currently operates on a dual-processor Honeywell 635 system with 160K words of core. Two DSS-170 storage devices provide 72 million words of disc storage. There are 160 ports available on the system, at speeds ranging from 110 to 2000 bps. Figure 2 shows the current hardware configuration.



Dartmouth Time-Sharing System
Honeywell 635 Hardware Configuration

Figure 2

DTSS--CAPACITY

Daily Throughput

DTSS processes more jobs in a single day than most computer centers process in an entire month. As many as 20,000 jobs have been processed by DTSS in a single 17-hour period. Note that 1% of these are long running jobs requiring several minutes of processor time. At Dartmouth 1000 different users log in for 3000 sessions each day. In an active month 5000 different persons use DTSS.

Wide User Base

There are 25,000 valid user numbers on the version of DTSS at Dartmouth. These users include 50 colleges and high schools distributed throughout seven eastern states and the Province of Quebec in Canada. DTSS also services a small number of commercial firms in the northern New Hampshire and Vermont areas. The system provides useful services to a wide spectrum of users. More than 90% of the 3000 undergraduates at Dartmouth use DTSS in the academic year.

Simultaneous Users

DTSS supports more simultaneous users than any other time-sharing system. Averaged over all the hours of system operation, the average number of simultaneous users is 50. Hardly a day goes by in which the peak number of simultaneous users is not greater than 100. The system has in fact on occasion handled as many as 137 users with response times to most commands remaining under ten seconds. The hardware installed at Dartmouth will allow up to 160 users if the demand requires this. DTSS software will allow about 250 users.

Resource Usage

On a typical day 1000 hours are logged by users of DTSS at terminals. Eight hours of processor time are used and billed. Nearly a half-million accesses to the disc storage systems are completed in the same period.

Cost Effectiveness

The cost effectiveness of DTSS is due largely to the capacity of the operating system to allocate the minimum number

of resources necessary to perform a given task. The software system was designed to provide service to 200 users; at each step of the design the question was asked -- "What would be the effect of 200 users on the system?"

DTSS--AVAILABLE LANGUAGES

BASIC

BASIC was developed at Dartmouth in 1964. Although it was originally designed with emphasis on ease of learning, it has continually been expanded to become DTSS BASIC -- a language including such features as

- data packing and unpacking
- direct access file support
- independent subprograms
- string handling
- interactive symbolic program debugging.

FORTRAN IV

FORTRAN IV on DTSS is second only to BASIC in popularity. DTSS FORTRAN IV contains several additional features which allow file access in a manner compatible with BASIC. A new FORTRAN language processing system is under development; it contains the full ANSI standard as a subset.

COBOL

A COBOL system has been developed which will allow rapid construction and debugging of COBOL programs from remote terminals. DTSS COBOL is full ANSI COBOL 68 with the exception of the segmenting feature. The COBOL compiler and run-time package have been verified using the Navy-prepared audit programs.

ALGOL 60

ALGOL 60 is also available on DTSS. The version available on DTSS follows "The Revised Report on the Algorithmic Language ALGOL 60" (Communications of the ACM, January 1963) with some commonly accepted exclusions. In addition DTSS ALGOL 60 contains

- double precision capability
- string handling facilities
- input/output for terminals and files
- chaining to successive ALGOL 60 programs.

ALGOL 68

ALGOL 68 is an extensible language of the ALGOL type, having features for the manipulation of structured data, strings of characters, bits, etc. Unlike PL/1, it is a safe language which insures at compile-time that an incorrect program cannot destroy data or sequencing information. ALGOL 68 is under development although some limited experimental use is possible.

DTRAC

DTRAC is the DTSS version of a published language for Text Reckoning and Compiling. It is essentially a character-handling language with a macro facility.

LISP 1.5

LISP 1.5 is an interpretive language for the manipulation of list structures. The DTSS LISP 1.5 interpreter most closely resembles that implemented at one time by the MIT Project MAC artificial intelligence group.

MIX

The MIX language system implements the ficticious computer used to illustrate computer operation by Donald Knuth in his series of books The Art of Computer Programming. A machine language interpreter, a debugger, and a symbolic assembler exist.

GMAP

GMAP is the assembly language for the Honeywell 635 which is the host computer for DTSS. This is a high-speed, two-pass macro assembler with excellent cross-reference features. Assemblers are also available for Honeywell's Datanet-30 Communications Computer (DMAP), Digital Equipment Corporation's PDP-9 (9MAP), Data General's NOVA (NMAP) and IMLAC Corporation's IMLAC (IMAP).

DDT

DDT is a system for debugging machine language programs produced by the GMAP assembler. DDT is highly developed, with selective breakpointing, symbolic patching, and file processing abilities. This technique was used to debug the entire DTSS software system.

Editors

A variety of editing systems are available to users of DTSS. One is oriented to process files of ASCII characters arranged in the line number order which most users prefer. Another editor operates on context, and a third allows the flexibility of regular string expressions and programs of edit commands.

DTSS--TIME-SHARING BATCH-PROCESSING CONTINUUM

Time-Sharing

DTSS is most widely known for its time-sharing capability. Indeed the first goal of DTSS was to minimize the effort required for a user at a terminal to compose and execute computer programs. Such interactive usage has expanded through the development of sophisticated data analysis packages, file-system maintenance procedures, and hardware maintenance aids.

Background

The background mode of operation is available for jobs which require long running times or other resources not normally available to interactive users. These resources would include the magnetic tape drives, the high-speed printer, etc. The user may write and debug a program in the interactive mode and then submit a background job description to schedule the job to be executed at the discretion of the Background System Monitor. The job specification statements are checked immediately upon submission of a job, so that simple job control statement errors are immediately brought to the attention of the user. Status reports are appended to the job description file as the work progresses so that a user at a terminal may determine, for example, if he should go to the computing center to pick up his output. This illustrates how the best features of foreground and background operation are made available to the DTSS user.

GCOS

The traditional operating system for the 600/6000 line is GCOS. Users of DTSS who also desire GCOS III services may obtain them through the GCOS III Subsystem Simulator which operates under the Background Monitor. The expanded error messages presented under this mode of operation are only one of the advantages of operating GCOS under DTSS.

Users may submit decks of cards in the traditional manner, or they may indicate a file which contains the GCOS III job. This mode of operation enables all of the capabilities of DTSS to be brought to bear on the problem of constructing and updating this job submission file. An important advantage is that most control-card errors are detected at the time of submission of a GCOS job. Another benefit is that reports may be appended incrementally to a file as job steps are completed, thus allowing

a user at a terminal to follow the progress of a GCOS III job through the system. A side benefit is the reduction in card handling by the operator; DTSS operates with only one operator.

DTSS--TERMINAL SUPPORT

Terminal Support

DTSS supports all popular time-sharing terminals. Software has been developed over a period of five years for the Datanet-30 communications controller which makes economical, reliable support possible for the following terminals:

Teletype 33/35	110 bps
Teletype 37	150 bps
IBM 2741 (EBCDIC)	134.5 bps
IBM 2741 (Corrsp)	134.5 bps
Terminet-300	300 bps
Memorex 1250	600 bps
Teletype Inktronic	1200 bps
Remote Batch Terminal	2000 bps

DTSS can operate with as many as 250 terminals; it has been tested successfully with 160 ports operational.

DTSS--APPLICATIONS

Applications

The prolific development of system services has been exceeded only by the development of applications services by the thousands of students, faculty, administrators, and others who number among the 25,000 active users of DTSS. The development of advanced applications programs has placed demands for expanded system services upon the DTSS development staff at Dartmouth. This peculiar symbiosis between applications development and DTSS development has resulted in a balanced, cost-effective resource available to the Dartmouth community.

For example, although most files on DTSS are ASCII, special packing and unpacking facilities were made available in BASIC to enable a survey summary system to be developed using random-access files. The IMPRESS (Interdisciplinary Machine Processing for Research and Education in the Social Sciences) application has won national acclaim for the facile manner in which hypotheses of social interaction may be tested against real data.

The easy use of foreground and background processing for data base management has encouraged the development of traditional data processing applications in the flexible environment of DTSS. Project management has been implemented on DTSS, and several management information systems have been put into use with good results.

Libraries of hundreds of applications programs are currently available on DTSS, and are constantly being expanded and augmented. In addition to programs, supporting curriculum materials for undergraduate education are prepared under the auspices of several projects underway at Dartmouth.

0. Introduction

This memorandum is intended to help a user of the Dartmouth Time-Sharing System (DTSS) sign on the system and then to interact with the computer in order to create and alter information stored in it.

The material presented in the first eight sections is an introduction to the use of the DTSS and provides all the beginning user needs, to know to sign onto the system, create new files and alter saved files, and use library programs. Section 9 gives a complete description of the system commands available to all DTSS users; section 10 explains the error messages you may receive while using the system commands; section 11 explains how to control the operations that may be performed on files you save; and section 12 contains a glossary of terms used. Note that the beginning user need not be concerned with sections 9 through 12 until he wants to know about some additional commands he may use, or until he is ready to investigate further the commands introduced in the first eight sections.

Note that this memorandum does not describe the BASIC language or any other programming language. The 6th edition of the BASIC manual, which describes Dartmouth Time-Sharing BASIC, is available to the Dartmouth College community from the Dartmouth Bookstore on Main Street in Hanover. Mail orders should be sent to University Press of New England, P. O. Box 979, Hanover, N. H. 03755. You may LIST the library file DOCUMENT*** for information regarding other available documentation (section 7 explains how to call up library files).

1. How to Call the Computer

Because of the variety of terminals and communications equipment, there are a number of ways to access the computer.

1. The eight Model 35 terminals closest to the front of the building in the public terminal room of the Kiewit Computation Center are wired directly to the computer; you only need to depress the ORIG button.

2. All other terminals must be connected through regular telephone lines. If the terminal has a data set connected to it, you must dial a number on the data set. The dialing procedure is described below; the phone numbers for different types of terminals are given on a supplement to this Technical Memorandum. A copy of this supplement may be found at the end of section 12.

   If the terminal has an acoustic coupler, dial the computer on the telephone beside the terminal, turn on the acoustic coupler and the terminal, and press the handset of the phone firmly into the coupler. The cord of the handset should be at the end of the coupler opposite the power cord.

   If you are uncertain what kind of terminal you are using, the following points may help.

   a. Many Model 33 Teletypes have an ON-OFF-LINE switch in the front right corner. Other identifying characteristics are the plastic cover and the exposed paper roll.

   b. Model 35 Teletypes have a metal case. A Model 35 Teletype is larger than a Model 33 Teletype and is generally not set on a table. The paper roll is enclosed.

   c. Model 37 Teletype terminals have upper and lower case characters and a row of 12 square and 2 rectangular buttons above the keyboard.

   D. The IBM 2741 Communications terminals, Friden, Novar, and Terminet terminals are clearly labeled as such.

Often detailed instructions for reaching the computer will be posted near your terminal. If you have difficulty signing on the system, dial (603) 646-3473 for help; if the terminal is faulty, please dial (603) 646-3253 to report the trouble.

2. How to Sign on the Computer

When you are connected to the computer, you will receive a sign-on message. When you see USER NUMBER--, type in your user number and push the RETURN key. The terminal will black out an area and ask for your password. Type in your password in upper-case letters (unless you have made special arrangements) and push RETURN. If you are a Dartmouth undergraduate student, your user number is your student ID number and your password is normally the first three letters of your last name.

NEW OR OLD-- will next appear on the terminal. You should type NEW if you wish to create and store new information and OLD if you want to retrieve information previously stored. (If you are using a Friden terminal, type FRIDEN, wait for the message READY, then type NEW or OLD.)

You will next be asked to supply a file name. A file is a collection of information stored in the computer. If you are creating a new file, you may type a name up to eight characters long; the name may contain uppercase and lowercase A through Z, numerals zero through nine, hyphen (-), and period (.). File names may not contain blanks. If you wish to work on a saved file, type the name used when the file was saved. You may type NEW or OLD followed by one space and then the name of the file on the same line if you wish.

You must push the RETURN key after each response you type; doing so tells the computer you have completed your response and it should examine what you have typed and act accordingly.

The following example illustrates the sign-on procedure. Information you type is underlined.

```
DARTMOUTH TIME-SHARING
LINE 110 ON AT 11:37  17 NOV 71, 074 USERS
DTSS TILL 20:00. LIST CCNEWS*** 11/05/71.
COMMUNICATIONS GAP??? RUN SUGGEST***.

USER NUMBER--431887
MMMMNMMM <-- PASSWORD
NEW OR OLD--NEW TRYOUT
READY
```

If you wish, you may type your password on the same line as your user number; in this case, type your user number, a comma, and then your password with no intervening blanks. On full duplex terminals (see FULL command in section 9) such as the model 33, 35, and 37 teletypes and Terminet terminals, the terminal will overprint your password, if necessary, so that others cannot read it. On half duplex terminals (see HALF

command in section 9) such as the Friden, IBM 2741, and NOVAR terminals, this overprinting does not occur. Thus, for your own protection, on half duplex terminals you should not type your user number and password on the same line.

If you would like to change your password, you may do so by contacting Nancy Broadhead at Kiewit, phone (603) 646-2643.

3. Entering Information

When the message READY appears, you may begin creating a new file or altering a saved file. Each line you type must begin with a line number, unless you are in the BUILD or DIRECT mode (see section nine) or unless what you type is one of the system commands listed in section nine. A line number consists of one or more decimal digits with no intervening characters. A line number ends with the first nondigit; the largest allowable line number in the programming language BASIC is 99999.

You may type any information whatsoever following a line number. If you make a mistake while entering your information, you may delete the last character typed:

1. on Model 33 and 35 Teletypes by typing a back-arrow (shift-oh).

2. on Model 33, 35 and Model 37 Teletypes, Terminet 300's and Friden terminals by typing CONTROL-Z (hold down the CTRL or CONTRL key while typing a Z).

3. on IBM 2741 and Novar terminals by typing a commercial at sign (@) followed by a Z.

You may delete all information on the line you are currently typing:

1. on Model 33, 35 and Model 37 Teletypes, Terminet 300's and Friden terminals by pressing CONTROL-X (hold down the CTRL or CONTRL key while typing an X).

2. on IBM 2741 and Novar terminals by typing a commercial at sign (@) followed by an X or by typing a backspace and immediately pressing the RETURN key.

You need not type the lines in numerical order; the computer arranges them. You may examine the contents of the file on which you are working by typing the command LIST. To correct mistakes in a particular line, type the line number of the line to be corrected followed by the correct information. To delete a line, type the line number of the line to be deleted and immediately push the RETURN key. You may begin listing the file at a particular number by typing LIST, then a blank, and the line number at which you wish the listing to begin.

To save a file for future use, type the command SAVE. If you are making alterations to a file previously saved, type the command REPLACE. You may change the name of your file by typing the command RENAME followed by a blank and the desired name.

If you wish to erase information previously stored type OLD, a space, and the name of the file you wish to erase. Then type the command UNSAVE. You may also simply type the command UNSAVE, a space, and then the name of the file on the same line. When you are finished you type BYE to let the computer know you are done.

The following example illustrates some of these procedures. (For a complete description of these and other commands, see section 9.) Information you type is underlined. The other printout is typed by the computer as responses to commands you have typed. Note the message READY, which means the computer has completed action on the last command you gave. After READY appears, you may enter information (usually line-numbered information) or may type another command (see section nine). If the message WHAT? appears, you have mistyped a command or have not begun your line with a line number.

```
NEW NEWFILE
READY

INFORMATION TYPED
WHAT?

200 INFORMATION TYPED INTO
210 THE COMPUTER MAY CONSIST
220 OF ANY CHARACTERS.
SAVE
READY

210 THE COMPUTER NEED NOT
215 BE LINES OF A PROGRAM
216 BUT MAY CONSIST
LIST

NEWFILE   30 SEP 71   11:35

200 INFORMATION TYPED INTO
210 THE COMPUTER NEED NOT
215 BE LINES OF A PROGRAM
216 BUT MAY CONSIST
220 OF ANY CHARACTERS.
READY

REPLACE
READY

BYE
```

In this example, the NEW command tells the computer that you want to create a new file that will be named NEWFILE. After the

READY message appears, you begin typing information. When you push the RETURN key, the computer tells you it did not recognize what you typed by saying WHAT?; in this case, you did not begin your line with a line number.

Three lines of textual material are next entered, and SAVE causes the computer to store the information for later use. Next line 210 is retyped, and 2 lines of new text are added between lines 210 and 220. Notice that when you list the file, the computer sorts the lines in increasing line-number order.

You type REPLACE so that the alterations you typed after you typed the SAVE command will be kept for future reference. Then you type BYE to let the computer know you are finished. The computer types a sign-off message; if necessary, you should turn your terminal and coupler off and hang up the phone after the sign-off message.

At some later time, you may wish to make further alterations to NEWFILE. Sign on to the system, and when NEW OR OLD-- appears, type OLD NEWFILE. You may then LIST the file to see what it looks like; it will be identical to the copy preserved the last time you worked with it. Make additions or corrections to your file by typing line-numbered information. If you then wish to preserve the modified version of NEWFILE, type REPLACE.

If the lines you type into the computer are statements of a program, you may type the command RUN to begin the execution of your program. For a complete description of the BASIC programming language, see the BASIC manual, sixth edition. This manual is available at the Dartmouth Bookstore for the college community; mail orders should be sent to University Press of New England, Box 979, Hanover, N. H. 03755.

Note: when you are entering information into the computer, it is advisable to SAVE and then REPLACE your file fairly often. Machine or terminal malfunctions may cause you to lose a lot of work if this precaution is not taken. (Remember that the latest alterations to your file are not preserved when you are disconnected from the computer unless you have typed SAVE or REPLACE.)

Complete descriptions of the system commands discussed above are available in section 9. Of particular interest to the beginning user are the commands BYE, CATALOG, EDIT, EXPLAIN, GOODBYE, LENGTH, LIST, NEW, OLD, RENAME, REPLACE, RUN, SAVE, and UNSAVE.

4. How to Interrupt the Computer

The null character will interrupt the listing of a file, the execution of a program, or any other activity. This is a good way to stop a program which, due to an error, would not otherwise terminate execution. The null character may be generated

1. on Model 33 and 35 Teletype terminals by typing CONTROL-SHIFT-P (hold down the CTRL and SHIFT keys and simultaneously type a P).

2. on Model 37 Teletype terminals by pressing the NULL key.

3. on IBM 2741 and Novar terminals by pressing the ATTN key.

The computer's response is STOP, READY. At this point you are now able to resume file building. The null character will always work.

The BREAK key or button (on Model 33 and 35 Teletypes and Friden terminals) will always interrupt the computer. Do not hold the BREAK key down for longer than about 1 second or you may be disconnected. On some terminals it may be necessary to depress the BRK-RLS light to re-establish communication after you have pressed the BREAK key.

Typing the character S will interrupt the computer when it is sending information to you.

When information is being sent from the computer to your terminal, you may cause it to skip up to about 250 characters by depressing CTRL-X (hold down the CTRL or CONTRL key and type an X).

When a program is being executed (in response to a RUN command), you may find out the accumulated computer time used for the program's execution by simply depressing the RETURN key when the terminal is not printing or awaiting input.

5. Some Limits on Your User Number

Your user number on the DTSS has associated with it information regarding your use of the DTSS: what hours of the day you may use the system and what activities you will be able to carry out. In particular, there is a limit on the amount of information you may save on the system, on the number of seconds you may use for each program run, and on the space each program uses when it is run.

To find out how much space the files you have saved are occupying, type the command CATALOG LENGTH. A heading giving your user number (preceded by the letters HD), the time and date, the total number of words occupied by the files you have saved, and the maximum number of words you are allowed to save will be printed. (A word is four characters). Under the header, the names of all files you have saved together with their lengths (in words) will be printed. Since the amount of information you may save on the system is limited, you will want to keep track of what files you have saved and their lengths so that you can remove unused files from the system (see section 6) to make room for other files. Under certain conditions, the maximum number of words you may save will be increased; you may apply for an increase in your storage maximum by coming to Kiewit or by phoning (603) 646-2643.

There is a limit on the number of seconds you may use for each program run or other activity. (The number of seconds used for a particular activity, such as a program run, is printed at the completion of the activity.) Unless you have made special arrangements, your limit per run is 32 seconds if you are a Dartmouth undergraduate or high school faculty member; 64 seconds if you are a graduate student or Dartmouth (or other college) faculty member; or 16 seconds if you are an elementary or high school user or non-Dartmouth undergraduate. There is no way at present that you may find out on-line the run-time limit on your user number. For this information, you may contact a staff member at Kiewit at the phone number given above.

There is also a limit on the amount of computer memory or space you may use for each program run; the limit for programs written in the BASIC programming language is 16,384 words. The system programs required to run a BASIC program occupy about 8,000 words, so you must write BASIC programs which use less than about 8,000 words. You may use the RUN SIZE command (see section 9) to see how much space a program run uses. There are various ways to economize on storage used by BASIC programs; see the 6th edition of the BASIC Manual, available from Univerisity Press of New England, Box 979, Hanover, N. H. 03755, for more information.

6. Automatic Removal of Unused Files

For the vast majority of users, FILES WHICH HAVE NOT BEEN USED FOR ABOUT A MONTH ARE REMOVED FROM THE SYSTEM. This procedure avoids the problems resulting from the proliferation of large numbers of unused files that would eventually use all the available space for storing files on the system.

The Computation Center maintains magnetic tape copies of stored files which can be used to recover files that have inadvertently been destroyed, either through a system error or by a user's mistake., THESE TAPES, however, DO NOT PROVIDE LONG-TERM PROTECTION; each user is responsible for obtaining his own "backup" copies of files that will not be used for a period of time.

There are several ways you may obtain copies of files to be stored outside the computer; if you want to use this information at a later date, it is easy to re-enter it. Paper tape, punched cards, and magnetic tape can be used for this purpose. Terminals with paper tape punches are available to nearly every user; some are located in Kiewit's public terminal room. Type

```
EXPLAIN PAPRTAPE
```

for instructions on how to punch paper tape. Punched cards may be used by members of Dartmouth College. Type

```
EXPLAIN BMC
```

for information on how to request a card deck of a saved file. Magnetic tapes and instructions for their use may be obtained in person or by mail from Kiewit; request Technical Memorandum TM008, the Background Users Manual, from the Kiewit Document Center, Kiewit Computation Center, Hanover, N. H. 03755, phone (603) 646-2643.

7. Accessing Library Programs

7.1 The Main Program Library

The Kiewit Computation Center maintains a library of programs (or other files, such as files of subprograms) which anyone with a valid user number on the Dartmouth Time-Sharing System may use. Descriptions of these programs are listed in TM010, A USER'S GUIDE TO THE DTSS PROGRAM LIBRARY, which is available for $2.00 from the Documents Center, Kiewit Computation Center, Hanover, N. H. 03755, phone (603) 646-2643. The files listed in TM010 are in the public domain; you may list them (except for certain data files and programs that are saved in a special form), modify them to suit your needs, and save them under your own catalog (provided you have sufficient storage).

You may obtain a list of all the program categories while you are using the system, if you like, by using the following procedure.

You type:	OLD DARTCAT***
Computer responds:	READY
You type:	LIST

A list of the categories and their descriptions will be printed on the terminal. For example, ANTHRO***, INVEST***, and STATIS*** are three of the categories listed. To obtain a list of the programs in any category follow the same procedure as above but substitute the name of the category for the file name DARTCAT***. For example, to obtain a list of the programs in the category STATIS***:

You type:	OLD STATIS***
Computer responds:	READY
You type:	LIST

To access individual programs follow the same procedure, typing the program name in place of the category name. LISTing the program will cause a brief description of the program as well as complete instructions for its use to be printed out on the terminal.

After the instructions have been printed, you may interrupt the terminal's printing as described in section four. Follow any special instructions, and, if the file is a program, type RUN to begin execution of the program.

A few programs in the library are stored in a special form (they are pre-compiled) to save both computer time and your time when you use the program. These programs may not be listed; when

you type LIST you will receive an appropriate message. Simply RUN these programs for instructions on their use.

7.2 Sublibraries

In addition to files which are saved in the main program library, there are a number of files saved in sublibraries, or subcatalogs, in the public program library. Some of these sublibraries are maintained by Kiewit and are used to group together collections of related files. Other sublibraries are used by courses, departments, and projects to bring together collections of programs useful for certain users. To find out the names of the current sublibraries, their subjects, and the people responsible for them, type

```
EXPLAIN SUBLIBS
```

The CATALOG command may be used to obtain information about some of the sublibraries (when you type EXPLAIN SUBLIBS, the sublibraries that allow this option are identified). For example, by typing

```
CATALOG OF SCODELIB***
```

you can obtain a list of all files saved in sublibrary SCODELIB***. Any of the options normally available with the CATALOG command may be used (see the CATALOG command in section 9).

To call up a particular file in a sublibrary, use the OLD command. For example, to access the file BRIDGE in sublibrary SCODELIB***, you should type

```
OLD SCODELIB***:BRIDGE
```

Files saved in sublibraries are not necessarily in the public domain; the person in charge of the sublibrary may restrict access to any file by specifying particular access permissions when he saves the file (see section 11). However, files saved in the sublibraries which are maintained by Kiewit are in the public domain. Some of the sublibraries maintained by Kiewit follow.

Name	Purpose
CHAINLIB***	Programs to which programs in the main program library or in other sublibraries chain.
DATALIB***	Data files used by library programs.

Name	Purpose
PLOTLIB***	Subprogram files of plotter software for the Timeshare Devices, Inc. plotter, standard terminals, and other plotting devices.
SCODELIB***	Source code for compiled programs in the main program library and in those sublibraries maintained by Kiewit.
SUBSLIB***	Files of subprograms used by library programs, or of general interest; the source code for compiled files in SUBSLIB*** is in SCODELIB***.

In general, there is no catalog available which lists descriptions of files in the sublibraries, although further information about a sublibrary may sometimes be obtained by contacting the person in charge of the sublibrary. Files of general interest which are saved in sublibraries maintained by Kiewit are listed in TM010.

For more information about sublibraries, including how to set up and maintain them, consult Technical Memorandum TM030, SUBLIBRARIES ON THE DTSS. Copies can be obtained from the Kiewit Document Center at the address given in section 7.1.

8. Sources of Information

8.1 On-Line Information

Much information is available to you through use of system commands.

You may call up the library file DOCUMENT*** (using the OLD command) as described in section 7 and LIST it for a complete list of available computer-related publications.

CCNEWS***, another library file updated regularly, contains announcements concerning new or revised programming procedures, equipment and schedule changes, Kiewit lecture series schedules, new publications, and other information of general interest. The date following CCNEWS*** in the sign-on message (see the example in section 2) indicates when CCNEWS*** was last updated. Listing CCNEWS*** is one of the most efficient ways of finding out new developments at the Center.

DTSSINFO*** is also a library file of general interest to users. It contains a list of files that give information about the DTSS. Included are frequently updated files giving rates for using the DTSS, system performance reports, instructions for using paper tape, and so on.

The library program SUGGEST*** provides a way for you to make suggestions and comments or to ask questions. When you RUN this program, it saves your message, which will be reviewed by a staff member. A response will be returned if you like, and attempts are made to correct any problems uncovered.

Typing the command EXPLAIN will produce a list of topics about which, using the EXPLAIN command, you may obtain more information. For example, typing

```
EXPLAIN LANGUAGE
```

will provide a list of programming languages available on the DTSS. Explanations of all the system commands discussed in section 9 are available through use of the EXPLAIN command. For example, typing

```
EXPLAIN LIST
```

provides a complete description of the LIST command.

8.2 People to Contact

Following is a list of references for computer-related problems you may encounter.

AREA	NAME	LOCATION	EXT.
General problems and advice	Terminal Room Assistants	Pub. Terminal Room, Kiewit	3473
	Stephen V. F. Waite	104 Kiewit	2643
Programming Assistance BASIC	Terminal Room Assistants	Pub. Terminal Room, Kiewit	3473
	Diane Mather	105 Kiewit	2643
	Stephen V. F. Waite	104 Kiewit	2643
Validating user numbers; granting special permissions	Nancy Broadhead	Secretarial area, Kiewit	2643
Supplying manuals	Jann Dalton	Kiewit basement	2643
Programming Assistance FORTRAN & GCOS	Stephen V. F. Waite	104 Kiewit	2643
	Mickey Spencer	107 Kiewit	2643
Use of TDI Plotter	Arthur Luehrmann	116 Kiewit	2643
Obtaining terminals	Leonard Cohen	109 Kiewit	2643
Developing curricular materials	Arthur Luehrmann	116 Kiewit	2643

In addition, if the terminal you are using is not functioning properly, please call (603) 646-3253 and supply the following information.

1. Your name.
2. The location of the terminal you are using.
3. The telephone number appearing on the data-set or telephone supplied for the terminal.
4. The nature of the problem.
5. The line number (given in second line of the sign-on message or printed if you type TTY).

9. System Commands

This section explains how to use the system commands available to all users of the Dartmouth Time-Sharing System. Some terms used need explanation before the description of certain commands will be clear. Any Technical Memoranda (TM's) mentioned are available from the Kiewit Documents Center, Kiewit Basement, phone (603) 646-2643. The BASIC Manual, sixth edition, is available from University Press of New England, Box 979, Hanover, New Hampshire, 03755.

9.1 Terminology

Under your catalog is the collection of all the files you have saved while working under your user number.

Your current file is the file which you are creating or altering at a given time. With a NEW command, a current file containing no information is created; subsequent to typing NEW, you may begin typing information into your current file. With an OLD command, a copy of the saved file referred to becomes your current file, and alterations are made to this copy; the saved copy of the file is not affected until you type REPLACE or UNSAVE.

Line-numbered alterations to the current file are not incorporated into it as soon as they are typed on the terminal. These alterations are stored in the alter file until the next command which causes the alter file to be sorted into the current file. Sorting the alter file into the current file consists of merging the lines of the alter file with the lines of the current file in ascending line-number order. If two or more lines of the alter file have the same line number, the one most recently typed will replace the others; a line consisting of only a line number will be treated as no line at all and will cause a line with that line number to be deleted from the current file.

When the alter file has been sorted into the current file, the alter file has no information in it, and the current file is the resulting combination of the sorted files. In general, if the execution of a command will involve your current file, this sorting will take place. IGNORE and LIST ALTERS are two commands which do not initiate a sort.

As you read the command descriptions, you may want to refer to the glossary in section twelve for more information.

9.2 Command Descriptions

Any command may be abbreviated to its first three characters. The information enclosed in < > in the descriptions which follow is information you supply. (Do not type the symbols < and > when typing the command.) Often this information is optional.

Here is a list of system commands grouped according to general function. An alphabetical list of commands together with a complete description of each command follows.

Elementary File Commands -- The following commands perform elementary file operations.

```
APPEND
BUILD
IGNORE
LIST
NEW
OLD
RENAME
REPLACE
SAVE
SCRATCH
SORT
UNSAVE
```

Program Execution Commands -- The following commands initiate the execution of a program.

```
BACKGROUND
COMPILE
DEBUG
GECOS
LINK
RUN
TEST
```

Editing Commands -- The following commands perform text editing functions.

```
EDIT
STRINGEDIT
TEXTEDIT
```

Terminal Mode Commands -- The following commands define the format of the data stream to or from your terminal.

DIRECT
FRIDEN
FULLDUPLEX
HALFDUPLEX
KEYBOARD
NFRIDEN
NPARITY
PARITY
TAPE

Miscellaneous Commands -- The following commands are not suitable for inclusion in any of the other groups.

BILL
BRIEF
BYE
CATALOG
EXPLAIN
GOODBYE
HELLO
JOIN
LENGTH
NBRIEF
PUNCH
SYSTEM
TTY
USERS

Experimental Commands -- The following commands execute the experimental versions of various systems.

XBACKGROUND
XBILL
XCATALOG
XEDIT
XEXPLAIN
XSTRINGEDIT
XTEST
XTXTEDIT

APPEND

Typing APPEND causes all alterations since your last NEW, OLD or REPLACE command to be appended to your current file. The alter file contains no information after the execution of this command, and neither the alter file nor the current file is sorted.

BACKGROUND

This command initiates the running of a program or other activities under the control of the Background Monitor. Background activities are carried out without the necessity of your terminal remaining connected. Permission of the computer center is required if this command is to be utilized. Your current file must be saved and must contain a description of the background activity, to be performed. The activity or job description is checked for errors, and appropriate diagnostic messages are printed if necessary. If the job description is correct, the message ***JOB ACCEPTED will be printed. As job steps are completed, status messages will be appended to the saved version of your current file. The Background Language is described in the BACKGROUND USER'S MANUAL, Technical Memorandum TM008.

BILL
BILL <month>

Typing BILL causes the printing of billing information about the current user number. Items included are terminal time and central processor time used, lines printed on high speed printer, cards punched, terminal time, Input/Output units, etc., together with the corresponding rates and totals (for most user numbers) for the current month.

You may specify a particular month for which you would like billing information; for example, BILL AUGUST may be typed. Monthly data are available for a one-year period; earliest data are for July, 1971.

BRIEF

Typing BRIEF will cause subsequent TIME and INPUT/OUTPUT messages and headers from program listings and runs to be suppressed. Also, an asterisk (*) is printed instead of READY to notify you that you may type line-numbered information or give another command. See the NBRIEF command.

BUILD

This command permits you to type in a file without line numbers. After you type BUILD, the computer will respond with the message SPEAK!. Subsequently you may type in (or enter from paper tape) lines which need not begin with line numbers. They will be appended directly to the current file rather than to the user file. You may terminate this mode of building a current

file by typing a line containing only a carriage return; after the computer responds READY you may give any command or type line-numbered information.

BYE

The BYE command causes you to be disconnected from the time-sharing system. The computer will print the time in the message OFF AT XX:XX, space up the paper, disconnect the telephone connection, and turn off the terminal if it is so equipped. On some terminals equipped with acoustic couplers, it is also necessary to hang up the telephone handset and turn off the terminal manually.

CATALOG
CATALOG <options>
CATALOG <options>;<file name 1> <file name 2> ...

The CATALOG command is used to initiate a printout of information describing files stored under your catalog. Typing the command CATALOG alone causes an alphabetical list of the names of your saved files to be printed. A header is also printed which gives summary information about the catalog and the files saved within it.

You may specify a list of options which cause additional information to be printed along with the file names. Normally, the file names are arranged in alphabetical order, but it is possible to suppress this sorting, to sort in reverse order, or to sort on other information (see below).

If you list particular file names following the option(s), then information will be printed only for the files specified. Individual options and file names may be separated by a blank or comma. A semicolon must separate the list of options and the list of file names.

The options appear in the following table and may be abbreviated to their first three characters.

Option	Explanation
ACCESS	Ways file may be used; see section 11 - for more information
ALL	NAME, LEN, ACC, DLM, DLU, and USE
ANAME	Prints ASCII directly; no octal conversion
CURRENT	Information requested becomes your current file and is not printed at your terminal
DATES	DLU and DLM

Option	Explanation
DLM	Date file was last modified
DLU	Date file was last used
EVERYTHING	NAME, LEN, OLE, ACC, OAC, DAT, USE, system file was saved under, STO, and OST
EXPLAIN	Explains options specified on same line
LENGTH	Lengths in words of files and catalog (a word is 4 characters in ASCII files)
NAME	Prints names of files stored in your catalog
NFILES	Prints catalog information only
NHEADER	Suppresses header
NSORT	Suppresses sorting of the catalog
OACCESSES	Octal representations of ACCESS
OF <catalog name>	Prints information about files in the catalog specified; for example, CATALOG LENGTH OF SCODELIB***;BRIDGE would print information about sublibrary SCODELIB*** of the public program library; for a list of sublibraries you may reference in the CATALOG command, type EXPLAIN SUBLIBS
OLENGTH	Octal representations of LENGTH
ONAME	Prints octal representations of file names
OSTORAGE	Octal representations of STORAGE
PRINTER	Prints 120 characters per line, not 75
RSORT <argument>	Sorts information in descending order according to the argument specified; arguments are same as those for SORT option
SORT <argument>	Sorts files before printing them according to the argument specified rather than by name; allowable arguments are ANAME, DLM, DLU, LENGTH, NAME, OLENGTH, ONAME, and USE; NAME is the default argument
STORAGE	Total length of all files in catalog printed in header
USE	Gives number of different days a file was used since the DLM

For a listing of all options type CATALOG EXPLAIN. For an explanation of any individual option type

CATALOG EXPLAIN <option>

For example, for an explanation of the option LENGTH, type

CATALOG EXPLAIN LENGTH

If a file name contains special characters such as lowercase letters or control characters, the file name will appear when you type CATALOG as two groups of 12 digits. These digits, taken in groups of three, are the octal numbers which correspond to

the 8 characters of the file name (if your file name is less than 8 characters long the remainder is filled with spaces, octal number 040). You can determine the characters in the file name by referring to the list of ASCII code numbers in section 8.3 of the BASIC manual or by running the library program CHARS***.

COMPILE

COMPILE <system name> <run time>

Typing COMPILE instructs the BASIC or other language processor to compile the current program; the resulting object code becomes the current file. The program is not executed, but subsequently may be saved for later execution. Using compiled programs eliminates repetition of the compilation phase of execution which can be time-consuming for very long programs. (However, use the RUN command until a program has been fully checked out and modification will be infrequent.) The name of the current file becomes .OBJECT. ; the uncompiled saved version of your program remains unchanged. In general, compiled programs that have been saved cannot be listed.

Unless you have given a SYSTEM command, BASIC will be assumed. You can also specify a system by supplying the optional <system name> field. The optional <run time> may be an integer number of seconds after which the compilation will be interrupted if incomplete. You may respond YES or NO when asked if the compilation should be continued. In the event that a compiler (such as BASIC) detects errors in the program, it will print error messages as appropriate and the current file will continue to be the source file (the program as you typed it) rather than becoming the compiled file.

DEBUG

You may type this command instead of RUN to begin the execution of a BASIC program. This command calls in a special part of the system, the DEBUGGER. After the BASIC program has been checked for correct format by the compiler the DEBUGGER begins operation. In many respects the DEBUGGER runs a BASIC program normally, but it allows the options of stopping the program at specified points, of printing values of variables and, in general, of tracing the progress of the execution. In addition, the DEBUGGER does not abort the program when an error condition occurs during the execution; you may change the value of variables and continue the run.

For complete information on the DEBUGGER, see the BASIC Manual, sixth edition.

DIRECT

This command sets the direct mode. In direct mode characters typed in from the keyboard or read in from paper tape are appended directly to the current file without change. In direct mode there are no special characters or actions associated with special characters. For example, typing a backarrow will not delete the previous character, and the backarrow will become part of the current file. If you do not enter any information for a period of 10 seconds, an exit from this mode occurs. You must then enter line-numbered information unless you retype DIRECT or enter the BUILD mode. (See the BUILD command.)

EDIT <function>

The EDIT command invokes a system which edits files on a line-by-line basis. The <function> specifies what operation is to be performed on the files. Type EDIT EXPLAIN for a complete list of available EDIT functions. For an explanation of the operation of a particular EDIT function type EDIT EXPLAIN <function>. A typical command is EDIT RESEQUENCE which conveniently renumbers the lines of the current file, beginning with 100 and incrementing by 10. The EDIT system is described in the DARTMOUTH EDIT manual, Technical Memorandum TM002.

EXPLAIN <list of topics>
EXPLAIN :CURRENT <list of topics> <:TTY> <list of topics>

Using the EXPLAIN command, you may obtain information on various aspects of the Dartmouth Time-Sharing System, such as the history of the system, the building, numbers to dial to access the system, and sources of information at Kiewit. Typing EXPLAIN will cause printing of a list of available topics. Typing EXPLAIN TOPICS will yield a list of topics and their corresponding subject matter. You may list a number of topics at once; separate the topics with a blank, a semicolon, or a comma. Your current file remains unchanged.

Typing EXPLAIN :CURRENT <list of topics> will cause the explanations associated with each topic specified to become your current file. These explanations are listed in the order used for the topic list. Any previous current file is erased; the explanations are not printed on the terminal.

It is possible to have explanations for some topics printed on your terminal and explanations for others replace your current file. If you type

EXPLAIN :CURRENT HELP MISTAKES :TTY DOCUMENT

the explanation for DOCUMENT will be printed on your terminal while the explanations for HELP and MISTAKES will become your current file.

FRIDEN

The FRIDEN command prepares the system to accept a Friden terminal. In this mode, a real time delay is invoked after a carriage-return character is output to the terminal to allow Friden carriages to keep up with output. This command also sets the mode of communication to half-duplex since all Friden terminals are half-duplex. To exit this mode, type the command NFRIDEN.

FULLDUPLEX

The FULLDUPLEX command turns on the feature which causes characters typed at a terminal to be transmitted back to the terminal. This is the proper mode of operation of a full-duplex terminal and is set automatically at sign-on at Model 33, 35, and 37 Teletypes.

GECOS

After approval of your application, you may use the command GECOS to submit a program to the GECOS-III batch processing simulator from a time-sharing terminal. To submit a job in this manner, you compose a GECOS job in your current file; each line of this job should correspond to one card of the deck you would submit to have the job run in a GECOS-III batch system. When you have typed GECOS, your job will run as soon as the necessary resources (core storage, central processor time, and peripherals) become available. Output from the job will be placed on the counter in the card equipment room at Kiewit. GECOS jobs may also be submitted in card form at the counter.

For more information, see Technical Memorandum TM023, titled HOW TO USE TIME-SHARING GECOS.

GOODBYE

The GOODBYE command causes you to be disconnected from the Time-Sharing System. This command is identical in operation to the BYE command.

HALFDUPLEX

The HALFDUPLEX command turns off the feature which causes characters typed at a terminal to be transmitted back to the terminal. This is the proper mode of operation for half-duplex terminals, such as the IBM 2741 Communications terminals, which are set automatically when you are connected to the computer.

HELLO

The HELLO command may be given when you want to terminate work under one user number and begin work under another user number at the same terminal. Typing HELLO allows you to do so without disconnecting the terminal from the Dartmouth Time-Sharing System.

IGNORE

The IGNORE command allows you to discard line-numbered alterations to your current file which you may have entered incorrectly. All alterations made since the last sort operation occurred are ignored. (In general, if the execution of a command will involve your current file, a sort will occur before the command is executed.) By typing LIST ALTERS you may see exactly what alterations the IGNORE command will discard.

JOIN <keyword>

The JOIN command allows several users to connect their terminals to the same program in BASIC or another language. The <keyword> is any series of up to eight characters. This command connects the terminal to the multiple-terminal conference named <keyword> if the conference has not yet been completed. See the description of the LINK command and MULTIPLE TERMINAL PROGRAMMING, Technical Memorandum TM009.

KEYBOARD

The KEYBOARD command is used to indicate that you will enter information from the keyboard rather than from the paper-tape reader. This is the normal mode of operation and is assumed when you sign on; in this mode the computer places a line-feed character after each carriage-return character typed from the keyboard. Also, it responds DELETED to each CTRL-X character after deleting the entire current line of type and deletes the previous character when CTRL-Z (or backarrow) is typed. See the TAPE command.

LENGTH

The length in words of your current file is printed. The number of characters in your current file is four times the number of words.

LINK <keyword>,<n>
LINK <keyword>, *

The <keyword> is any series of up to eight characters and <n> is some integer number greater than one and less than eleven. By typing "LINK <keyword>,<n>", you initiate an <n>-terminal conference named <keyword>.

The option * may be used in place of a specific number of terminals to indicate that the conference can support a variable number of conferees. The conference will be initiated immediately, and one to nine additional users may join the conference at any time. In addition, any user who leaves the conference (by pressing the BREAK key, for example), except for the user who typed the LINK command, may rejoin it by retyping JOIN <keyword>.

For more information, see MULTIPLE TERMINAL PROGRAMMING, Technical Memorandum TM009, and also the JOIN command in this memorandum.

LIST
LIST <argument> <line number>

Typing LIST will cause the current file to be printed on the terminal. If you type LIST with no arguments, then the alter file is sorted into the current file and the resulting current file is printed.

The optional <argument> field may be CURRENT or ALTERS. If the argument is CURRENT your current file is listed without sorting your alter file into it. If ALTERS is specified the alter file is sorted and listed, but lines containing only line numbers are not deleted.

If <line number> is greater than any line in the file, the last line of the file is printed. Otherwise, the list of the file referenced will begin at that integer line number or immediately thereafter if no such line number appears in the file.

If LISTNH or LISN is substituted for LIST, the file is printed without a heading.

NBRIEF

Typing NBRIEF terminates brief mode operation and returns your terminal to the normal mode. This will cause TIME and INPUT/OUTPUT messages and headers from program listings and runs to be printed as usual, and READY is printed instead of the brief mode asterisk. Also, typing NBRIEF causes all system usage charges accumulated under brief mode to be printed. For more information, see the BRIEF command.

NEW <file name>

The NEW command is given to begin the creation of a new current file. If you have made alterations to your previous current file but have not saved them they will be lost. The <file name> may consist of one to eight characters including lowercase and uppercase letters A through Z, digits zero through nine, period (.), and minus sign (-). If the file name is not given in the NEW command, it will be asked for immediately.

NFRIDEN

The NFRIDEN command suppresses the carriage-return delay and horizontal-tab delay necessary for operation with the slow carriage-return function of the Friden terminal. Additionally the mode is set for full-duplex operation. See the FRIDEN command and the FULLDUPLEX command.

NPARITY

The NPARITY command suppresses the normal even or odd parity setting of each character transmitted to a terminal. (This setting varies with the terminal model.) In this mode each character is transmitted exactly as generated by the program or command causing information to be transmitted to the terminal. This command is only used for specialized terminals such as the Calcomp plotter or for punching tapes for unusual machines such as numerically-controlled machine-tools or automatic typesetters. See the PARITY command.

OLD <file name>
OLD <file name>,<password>

The OLD command is used to access a file which has been previously saved. If the <file name> is not given, you will be

asked to supply it. If the command is successful, the specified file will become your current file and any previous current file or alter file will be lost unless already saved.

If the specified file exists, you will be allowed to access it with the permissions permitted without a password. If no permissions are available without a password, you will be asked to supply one; if the password is correctly specified, you will be given those permissions available with a password. If no accesses are available either with or without a password or if you supply the wrong password, you will be denied access to the file. If you follow the <file name> with a comma and a <password> and the file name and password are correctly specified, you will be given those accesses available with a password. For more information on file accesses, see section 11.

OLD <file name>***
OLD <file name>***,<password>

This form of the OLD command is used to access a file saved in the public library of files in the Dartmouth Time-Sharing System. In all other respects it is identical in function to the previously described OLD command.

OLD <user number>:<file name>
OLD <user number>:<file name>,<password>

This form of the OLD command may be used to access a file belonging to another user and make it your current file. The <file name> refers to a file saved in the catalog <user number>. A password may be supplied in the OLD command, separated from the file name by a comma. If a password is required but not supplied in the OLD command, a request for the password will be made. For this command to be successful, the specified user number must be valid on the Dartmouth Time-Sharing System. In addition, the file must have been saved with read and public access or with read and group access if you are in the same group of 1000 user numbers. See the SAVE command.

OLD <sublibrary name>***:<file name>
OLD <sublibrary name>***:<file name>,<password>
OLD <sublibrary name>***,<password>:<file name>
OLD <sublibrary name>***,<password>:<file name>,<password>

This form of the OLD command is used to access a file saved in a sublibrary, which is itself saved in the public library of files. Such sublibraries can be used by courses, departments, or projects to group together a large collection of related programs

(see Technical Memorandum TM030 entitled SUBLIBRARIES ON THE DTSS). In all other respects this command is identical in function to the preceding OLD command. The passwords for the sublibrary will be requested only if required; the same statement applies to the programs.

PARITY

The PARITY command causes parity of characters transmitted to a terminal to be even or odd, according to the particular type of terminal. As an error check, some terminals, such as Model 37 Teletypes, require this parity setting for proper functioning. Model 33, 35, and 37 Teletypes accept even parity. The PDP-9 interface requires odd parity; the IBM model 2741 terminal uses six data bits plus a parity bit. This mode of operation is normal. See the NPARITY command.

PUNCH
PUNCH <argument>,<line number>

The PUNCH command is used to prepare paper-tape copies of the current file. It operates identically with LISTNH (or LISTN) except that rub-out characters are punched preceding and following the file. In addition, control characters are generated which turn the punch on prior to punching the file and turn the punch off subsequently. A control character is also punched on the tape which will cause the paper-tape reader to turn off at the end of the tape. These control characters are effective only on terminals which have the necessary features to permit the punch and reader to be turned on and off automatically. It is possible to prepare paper tapes on other terminals equipped with a paper tape punch by turning the punch on and off manually. For instructions on punching paper tape, type EXPLAIN PAPRTAPE.

RENAME <file name>

The name used to identify the current file may be changed with a RENAME command. The new name is specified as <file name>; it may consist of from one to eight characters, including uppercase and lowercase A through Z, the digits zero through nine, period (.) and minus sign (-). If the new name is not specified it will be asked for immediately.

REPLACE
REPLACE <file name>,<password>

The REPLACE command is used to replace a saved version of a file with the current file. If no <file name> is specified, the name of the current file will indicate the file to be replaced. If a <file name> is specified, the current file will replace the file named; the name of the current file will not change. The comma and password are optional. If a password is required to allow the replacement of the file and it is not supplied, a request for the password will be made. See the SAVE command.

RUN
RUN <system name> <run time>

The RUN command initiates the execution of the current program. The program will be assumed to be a BASIC program unless you have specified another system with the SYSTEM command or explicitly name a system in the optional <system name> field. Unless you have special privileges, the only systems which will be accepted are listed in the following table.

ALGOL	Dartmouth ALGOL; see TM021
ALGOL68	Under development
APL	Under development
BASIC	Dartmouth BASIC; see BASIC 6th edition
COBOL	Dartmouth COBOL; see TM029
DMAP	DATANET-30 Macro Assembly Program
FORTRAN	Dartmouth FORTRAN; see Joseph & Chambers, Ltd. publication available at Kiewit
GEFORT	GE FORTRAN; see ISD publication #802209D
GMAP	Dartmouth GE Macro Assembly Program; section 21 of DTSS Programming Reference Manual
LISP	Dartmouth LISP; see TM017
MIX	See Knuth, Fundamental Algorithms & TM013
OBASIC	Old Dartmouth BASIC; see BASIC Fifth Edition
SNOBOL	Under development
9MAP	Dartmouth PDP-9 Macro Assembly Program

"TM" stands for Technical Memorandum; the BASIC Manual, fifth edition, and all TM's may be obtained from the Documents Supervisor, Kiewit Basement, phone (603) 646-2643. The sixth edition of the BASIC manual is available to the college community from the Dartmouth Bookstore; mail orders should be sent to University Press of New England, Box 979, Hanover, N. H. 03755. The manual on GE Fortran is available from General Electric Co., Information Services Department, 7735 Old Georgetown Road, Bethesda, Maryland, 20014. See Miss Handy, phone (603) 646-2643,

at Kiewit regarding section 21 of the DTSS Programming Reference Manual.

An integer number may optionally be specified as the <run time> field. If the program attempts to run for a longer number of seconds you will be asked whether execution of the program should continue. You should answer YES or NO. Each user has a maximum amount of time he can use in a single run (see section 5).

RUN SIZE
RUN <system name> <run time> SIZE

If you are running a BASIC or FORTRAN program, this command causes the amount of core memory your program is occupying at the time the program terminates to be printed. If the program does not get beyond the compilation stage, core size will not be printed. In all other respects, this command is identical to the preceding RUN command. The RUN SIZE command is useful because it allows you to find out how your programs are developing in relation to your core size limit (normally 16,384 words), and it can help you write programs which make the most economical use of the system's core memory.

SAVE
SAVE <file name>,<password>;<accesses>

When you type SAVE the current file is saved in your catalog. If the SAVE command is followed by one or more blanks followed by a name, the file will be saved under that name; the name of the current file does not change.

It is also possible to save a file with a password specified by typing a comma after the command SAVE followed by the password. In this event, no accesses will be granted on a subsequent OLD command unless the proper password is specified. Accesses can be granted selectively; see section 11 for more information.

Note that if no password is specified for a file, it actually has a password of eight blanks.

SCRATCH

This command erases the current version of your file. After you type SCRATCH, both your current file and alter file contain no information. Saved files are not affected by this command and the current file name is not changed.

SCRATCH <file name>,<password>

This form of the SCRATCH command is used to discard the contents of a file saved in your catalog; the file remains saved but its length is set to zero. The current file is not affected. If a password is required in order to allow the saved file to be scratched and the password is not supplied, a request for it will be made.

SORT

This command may be used to sort a current file into ascending line-number order. If two lines have the same line number, the first will be ignored, and any lines without line numbers will be deleted. This command may be useful when a current file was created out of line-number order, for example, by using the BUILD command. Use this command with caution.

STRINGEDIT <function>

This command invokes the String Editor which may be used to edit files on a character-by-character basis. The <function> specifies the type of editing to be performed. The String Editor is described in Technical Memorandum TM003.

SYSTEM <system name>

This command may be used to specify a system other than your current system. Unless another system is specified in a RUN command or SYSTEM command, BASIC is assumed. Acceptable <system name> arguments for users without special privileges are those found in the description of the RUN command.

TAPE

This command is given to indicate you will enter information from the paper-tape reader rather than from the keyboard. In this mode, rubouts are ignored and a line-feed is automatically inserted after any carriage-return character which is not followed immediately by a line-feed, although the inserted line-feed is not printed on your terminal. Carriage-returns which are immediately followed by a line-feed are not modified. Also, typing CTRL-X causes the current line of input to be deleted, but the DELETED message is not printed on your terminal. As when you enter information from the keyboard, a CTRL-Z entered in the TAPE mode deletes the previous character. To exit this mode type KEYBOARD. (See the KEYBOARD command.)

TEST

This command is used to aid the student in testing programs written while he is learning to solve certain problems using the computer. The name of the program and the names of the variables used in the program must conform to the instructions issued by the instructor or the BASICT*** programs. See Chapter 6 of the 6th Edition of the BASIC Manual.

TEXTEDIT <argument list>

This command invokes the Text Editor system which allows the editing of files on a character-by-character basis. The Text Editor is easier to use than the String Editor, but it is more limited and requires much more computer time when used with large files. The <argument list> specifies which operations are to be performed. See Technical Memorandum TM004.

TTY

Your terminal line number, current system, user number, and current file name are printed on the terminal.

UNSAVE
UNSAVE <file name>,<password>

This command may be used to remove a file from your catalog. The current file is unaffected. If <file name> is not specified, the name of the current file specifies the file to be removed. If <file name> is specified, then that file will be removed. Whether or not a password is supplied depends on the accesses specified when the file was saved. (See the SAVE command.) If a password is required to allow the removal of the file and it is not supplied in the UNSAVE command, it will be asked for immediately.

USERS

The number of users currently on the Dartmouth Time-Sharing System is printed on your terminal.

```
XBACKGROUND
XBILL
XCATALOG <options>
XEDIT <argument list>
XEXPLAIN
XSTRINGEDIT <argument list>
XTXTEDIT <argument list>
```

These commands are used in the same manner as their counterparts: BACKGROUND, BILL, CATALOG, EDIT, EXPLAIN, STRINGEDIT, and, TEXTEDIT. An experimental version of the respective systems is used.

```
XTEST
XTEST <run time>
```

This command is used by the programmer developing a TEACH test program. With XTEST, a test may be carried out while the TEACH test program is in his catalog rather than in the official TEACH program library. An integer number may optionally be specified as the <run time> field. If the program attempts to run for a longer number of seconds, then the user will be asked whether testing should continue; he should answer YES or NO. See chapter six in the sixth edition of the BASIC Manual.

9.3 Multiple Commands

As you are learning the system commands, you should type each command on a line by itself. However, as you become familiar with the effects of the commands, you may want to type more than one command on a line. Three examples of multiple commands on a single line follow.

```
.OLD FTBALL***.RUN
#OLD FILEX#EDI RESEQUENCE#REPLACE
/NEW FILEY/BUILD
```

Beginning a command line with a delimiter (any character except a number, space, or a letter) signals the computer that a sequence of commands will be typed on that line. Several commands may then be typed on the line, each command separated from the next by the delimiter with which the line was begun. When you push RETURN at the end of the command line, the computer begins executing the commands in the order in which they were typed.

If an error condition occurs during the execution of the command sequence, the computer will stop execution of the command sequence and print an error message. You may stop the execution

of a command sequence at any time by following the appropriate procedure for interrupting the computer discussed in section four.

After executing a command sequence or some of the commands of the sequence, the computer prints the number of commands executed, a right parenthesis ")", and then the usual READY message. If all commands were successfully executed, the number preceding the READY message simply equals the number of commands that you typed on your command line. If an error condition occurs or if you stop execution of the command sequence, the number preceding the READY message tells you how many commands were successfully executed before execution was stopped.

10. System Messages

The following is a list of messages you may receive after typing one of the commands listed in section nine.

"<file name>" CANNOT BE SORTED; TYPE 'EXPLAIN COMSORT'

You have typed a command which requires that your latest line-numbered alterations be sorted into your current file. The sort cannot be performed, however, because your current file is a compiled program. A file can become designated as a compiled program in the following ways.

1. by being compiled from a source program via the COMPILE command. If the compiled program produced by the COMPILE command is then saved, it will retain its designation as a compiled program when accessed via subsequent OLD commands.

2. by specifying the compile access permission when saving the file (see section 11).

If you have incorrectly designated a file as a compiled program, you will be unable to modify it until you type the following sequence of commands.

/OLD <file name>/UNSAVE/SAVE;007601007601

This command sequence explicitly tells the computer that the compile designation is to be removed (the last 12 characters in the above command sequence are digits; do not use the letter "oh"; the characters are zeros.)

In any case, immediately prior to printing the above error message, the computer discards the alterations you typed in following your last OLD, SAVE, or COMPILE command.

"<name of sublibrary>" IS A LIBRARY; TYPE 'EXPLAIN OLD'.

Using an OLD command, you have referenced a sublibrary in the public library of files instead of a file. You must follow the sublibrary name with a colon (:) and then the name of the file you wish to access. See the fourth form of the OLD command in section nine.

"<system name>" IS BEING MODIFIED; TRY AGAIN IN A FEW MINUTES.

The system you have referenced in your RUN command is being modified. Retype the command in a few minutes.

"<file name>" IS BEING USED; TRY AGAIN LATER.

You are trying to access a file which someone is altering or which someone has accessed with permissions enough to alter. For example, someone may be running a program which has obtained enough permissions to write into the file you wish to use. Wait a few minutes and try again.

"<file name>" IS EXCESSIVELY LONG; TYPE 'EXPLAIN LONGFILE'.

This message will occur under any of the following conditions:

1. You typed APPEND, BUILD, RENAME, REPLACE, SAVE, SCRATCH, SORT, or OLD <user number>:<file name>, and your file is larger than 256,000 words (or 1,024,000 characters).

2. You typed LIST <line number> and there are more than 256,000 words of information preceding that <line number> in your file.

3. You were in the BUILD or DIRECT mode and you increased the size of your file so that it is now larger than 256,000 words.

In the first two cases, you will have to break your file into smaller segments (see Technical Memorandum 2 which describes Dartmouth Edit or TM003, the Dartmouth String Editor, available from the Document Center, Kiewit basement) and work with these shortened portions of your file.

In the third case, you may type REPLACE or SAVE and your alterations will be preserved.

If you have further questions, call Kiewit (603) 646-2643 for more help.

"<what you typed>" IS NOT A LEGAL SYSTEM

You have referenced something in a SYSTEM or RUN command which is not in the list of legal systems given in the

description of the RUN command in section nine. Check spelling and try again.

"<what you typed>" IS NOT A LIBRARY; TYPE 'EXPLAIN OLD'.

You have typed an OLD command of the form OLD <sublibrary name>***:<file name> but what you supplied for <sublibrary name> is a file in the public library of files rather than a sublibrary. You may have the position of the sublibrary name and file name reversed. See the fourth form of the OLD command in section nine.

"<what you typed>" IS NOT A VALID USER NUMBER.

You have typed an OLD command of the form OLD <user number>:<file name> and what you supplied for <user number> is not a valid user number on the Dartmouth Time-Sharing System. You may have the position of the user number and file name reversed. See the third form of the OLD command given in section nine.

"<file name>" IS NOT AVAILABLE; TYPE 'EXPLAIN NOTAVAIL'

You have typed a command of the form

OLD <user number>:<file name>

and the file specified by <file name> is unavailable for one of the following reasons.

1. The file is not saved in the specified user catalog.
2. The file is saved in the specified user catalog, but is saved with neither group nor public permissions. (For further information on permissions for saved files, see section 11.)
3. The file is saved in the specified user catalog with group or public permission available provided the proper password is specified. You either did not specify a password in your OLD command, or you specified an incorrect password.
4. The user who owns the file is currently modifying it via a program or via a REPLACE, SCRATCH, or UNSAVE command.
5. The specified user catalog contains more than twice the amount of file storage that it should.

"<file name>" IS NOT COMPILED PROPERLY; RECOMPILE IT.

You have typed RUN and the program to be executed is saved with compile permission but is not interpretable as compiled code by the system you are using. A previously compiled file may have inadvertently been changed, or you may have mistakenly saved a non-compiled file with compile permission. Call up your source file (the program as you typed it) and recompile it, or change the accesses on the compiled file by using the procedure given in the description of the first error message in this section.

"<what you typed>" IS NOT SAVED

The file you are attempting to REPLACE, UNSAVE, SCRATCH, or access through an OLD command is not saved. Check the spelling of the file name. See the SAVE command in section nine.

"<file name>" IS OUT OF ORDER; TYPE 'EXPLAIN ORDER'.

You have given a command which requires that your latest alterations be sorted into your current file, but your current file is not in increasing line-number order, or contains lines without line numbers. This may have happened when you were altering the file under the BUILD or DIRECT mode, through use of an EDIT command, through a program, or by repeatedly running a program which uses the BACKGROUND system.

The IGNORE, APPEND, and SORT commands will allow you to continue file building. You may examine your current file before typing one of these commands by typing LIST CURRENT <line number>. (The <line number> is optional.) You may examine your latest alterations by typing LIST ALTERS <line number>.

If you type IGNORE, all alterations you have typed in since the last command which erased your alter file (such as a NEW, OLD, LIST, or REPLACE command) will be lost and you will have to retype the alterations.

If you type APPEND, all alterations since your last NEW, OLD, or REPLACE command will be appended to your current file, and the alter file will be scratched. The resulting current file is not sorted.

If you type SORT, alterations since your last NEW, OLD, or REPLACE command will be merged with the current file and the resulting file will be sorted. WARNING: If you type sort all lines without line numbers will be deleted!

If you need further advice, you may call (603) 646-3473 or (603) 646-2643.

"<file name>" MAY NOT BE LISTED; TYPE 'EXPLAIN LISTPERM'.

To list a file, LIST permission must be available on the file. As you referenced the file in your OLD command, LIST permission was not granted. You may be attempting to LIST a compiled program. (Remember that compiled files are automatically saved without list permission.) See section 11 for more information.

"<file name>" MAY NOT BE SAVED; TYPE 'EXPLAIN SAVEPERM'.

Using an OLD command, you have accessed a file without SAVE permission and have then attempted to save the file with a SAVE or REPLACE command. The file you called may have been saved with certain access permissions available with a password and possibly different permissions available if a password is not supplied. If the file is saved in your user number, you may check passworded and nonpassworded accesses on the file by typing CATALOG ACCESS: <file name>. Passworded accesses may be obtained by supplying the password along with the file name in the OLD command. See the OLD and SAVE commands described in section nine. For a description of how to change accesses on a file, see section 11.

<list of permissions> PERMISSION(S) NOT AVAILABLE FOR "<file name>"

To carry out the command issued, the permissions listed are needed in addition to any permissions already available for this file. To find out the permissions (or accesses) on a file saved in your catalog, type CATALOG ACCESS: <file name>. To find out what permissions are needed to perform various operations on a file (such as LIST, UNSAVE, SAVE, RENAME, SCRATCH, etc.) and to obtain information on how to change permissions on a saved file, see section 11.

"<file name>" PREVIOUSLY SAVED; TYPE 'EXPLAIN SAVREP'.

You have typed SAVE and information is already saved in your catalog with the same name as that of your current file. This message prevents you from replacing files by mistake simply because your current file name happens to be the same as the name of a saved file. If you wish to replace the saved file, type REPLACE. If you wish to save the file with another name, type

RENAME <new file name> and then type SAVE.

Alternatively, you can type REPLACE <file name>, which replaces the saved copy of <file name> in your catalog with your current file but does not change the name of your current file.

You may also type SAVE <file name>. This command will save a copy of your current file under the name <file name> and will not change the name of your current file.

Note that the SAVE command which caused this message has not been executed. See the SAVE, REPLACE, and RENAME commands in section 9.

"<file name>" SAVED UNDER SYSTEM "<system name>"

This message appears in response to an OLD command. The system named was your current system when you saved the file named, but you are now working under a different system.

SIMDUMP<6 digit number>*<date>********<time>

This message indicates a malfunction in one of the programs which operates the system. You are not at fault. Please give or mail the terminal paper containing your work immmediately preceding this message to a Kiewit staff member.

BACKGROUND PROGRAM "<file name>" MUST BE SAVED

Any alterations you have typed since your last OLD, NEW, or REPLACE command must be incorporated into a saved version of your current file. Type REPLACE if the file was previously saved; otherwise, type SAVE. Then retype BACK.

BACKGROUND USAGE NOT ALLOWED; TYPE 'EXPLAIN BACKUSE'.

You have typed BACK and your user number is not validated to use the background system. Call (603) 646-2643 to find out if you may get permission to use this system. Technical Memorandum 8, available from the Document Center in Kiewit basement, describes use of the BACKGROUND system.

CAN'T EXPLAIN "<what you typed>"

You have asked for information on a topic for which no entry has been made in the computer. For a list of available topics

for the EXPLAIN command, type EXPLAIN. The topic you supplied which generated this message has been stored for later review by a Kiewit staff member. If information on this topic is thought to be of general interest (and computer-related) an explanation will be prepared. In the meantime, you may call 646-3473 or 646-2643 for assistance. The area code for Hanover is 603.

COMMAND TOO LONG; RETYPE IT.

You have typed a command line longer than 128 characters.

COMPILATION SOURCE FILE "<file name>" MUST BE SAVED

Alterations since your last OLD, NEW, or REPLACE command must be incorporated into a saved copy of your current file. Note that you cannot modify your current file, save it under another name using SAVE <file name> and then issue a COMPILE command for your current file. The saved copy of your current file must be the same as your current file. Type SAVE or REPLACE then retype COMPILE.

COMPILATION SUCCESSFUL

This message tells you the source code (the program as you typed it) of your current file has been compiled successfully. Your current file is now the object or compiled code; .OBJECT. is now the current file name. The saved copy of your source code remains unchanged.

COMPILATION UNSUCCESSFUL

This message will be preceded by detailed messages telling you why your program was not compiled. Your current file is still your source program. Correct your errors, type REPLACE, and then retype COMPILE.

COMPUTER HARDWARE ERROR; TYPE 'EXPLAIN HARDERROR'.

An error has occurred in the computer equipment through no fault of yours. The nature of the malfunction has been recorded automatically at the computer center for later review by a Kiewit staff member. You may retry the command but it is possible that it will not be executed successfully until equipment is repaired or replaced. If you still get the same message after several tries, you will have to call up the saved version of your current file (OLD command) or begin a new current file (NEW command). In

either case, all alterations since your last NEW, OLD, or REPLACE command are lost.

COMPUTER STORAGE EXHAUSTED; TYPE 'EXPLAIN NOSTORAGE'.

There is not a big enough block of storage in the computer to perform the command you issued. You can try breaking your file into smaller parts (see Technical Memorandum 2, Dartmouth Edit, or TM003, The Dartmouth String Editor, available in Kiewit basement at the Document Center) and then proceed. If this fails, you may call 646-3473 or 646-2643 for assistance. The area code for Hanover is 603.

COMPUTER TEMPORARILY OVERLOADED; TYPE 'EXPLAIN OVERLOAD'

You have typed a command that involves the execution of a system such as the BASIC compiler or the TEXT editor. Temporarily, however, it is not possible to bring the system into execution, for one of the following reasons:

1. The computer currently has more than 200 systems in execution.

2. There is not enough room on either the drum or disc storage units to store a copy of the system to be executed.

This situation should occur very infrequently and should be temporary in nature. Retype your command.

CONFERENCE "<keyword>" DOES NOT EXIST

You have typed a JOIN <keyword> command and no LINK command has been typed at another terminal using the same <keyword>. See JOIN and LINK commands in section nine.

CONFERENCE "<keyword>" HAS TERMINATED

The multiple-terminal conference of which you were a part has been discontinued. Your terminal has returned to its normal unlinked condition, and you may type any command or resume file building. See the LINK and JOIN commands in section nine.

CONFERENCE "<keyword>" IS ALREADY COMPLETE

You have typed JOIN <keyword> and a multiple-terminal conference using the same keyword exists but is already complete; that is, someone else has typed LINK <keyword>,N (using the same keyword you did) and N other persons are already in the conference.

CURRENT FILE "<file name>" IS EMPTY

You have typed a command which requires a non-empty current file (such as LIST OR PUNCH) and there is no information in your current file.

EMPTY ALTER FILE

You have typed LIST ALTERS and there is no information in your alter file. See the LIST command in section nine.

EXCESSIVE ALTERATIONS; TYPE 'EXPLAIN LONGALTERS'.

You are either typing line-numbered alterations to your current file or reading information into the system from paper tape, and you have entered more than 8192 words (or 32768 characters) since your last SAVE or REPLACE command. If you are entering information from paper tape, type BUILD and resume reading. Otherwise, type SAVE or REPLACE and continue file building. Note: you should rarely get this message because you should type SAVE and then REPLACE often when building or altering files.

FORMAT ERROR; TYPE 'EXPLAIN FORMAT'.

You have typed a command and have used improper punctuation or have the arguments or options following the command in the wrong order. Another common error is the inclusion of spaces or illegal characters in file names. Check the command formats given in section nine.

GECOS USAGE NOT ALLOWED; TYPE 'EXPLAIN GECOS'.

Before the system will allow you to submit a GECOS job from a terminal under time-sharing, you must have your application for such a privilege approved. You may make application for GECOS usage at Kiewit.

GROUP CATALOG MAX EXCEEDED

There is a maximum placed on the total number of words that all users in a group of 1000 user numbers may store in their catalogs. The storage maximum for the group to which your user number belongs has been exceeded; please call (603) 646-2643 to report the problem.

ILLEGAL ACCESSES SPECIFIED; SAVE NOT PERFORMED.

In your SAVE command, you have improperly represented the accesses with which you wish your file to be saved. See section 11 for more information.

ILLEGAL FILE NAME CHARACTER; TYPE 'EXPLAIN FILENAME'.

In a NEW or RENAME command, the filename you specified contained some character other than uppercase and lowercase A through Z, numerals zero through nine, hyphen(-), and period(.). Passwords for files may contain any characters except space( ), comma(,), semicolon(;), and colon(:).

INITIAL TIME LIMIT EXCEEDED; CONTINUE?

You have typed RUN <time limit> and the program has executed for the specified number of seconds. If you answer YES, program execution will continue; if you answer NO, the program run will terminate.

Y WORD "<keyword>" IS ALREADY IN USE; CHOOSE ANOTHER KEY WORD.

You have typed LINK <keyword>,<N> to initiate a multiple-terminal conference, but someone else has already initiated a conference using the same keyword. Retype the command with a new <keyword>. See the LINK and JOIN commands in section nine.

LIBRARY "<what you typed>" DOES NOT EXIST

You have typed OLD <what you typed>***,<file name> and <what you typed> is not a sublibrary in the public library of files. Check spelling and retry the command.

LINE TOO LONG; RETYPE IT

You have typed a line consisting of more than 251 characters.

PARITY ERROR IN ABOVE LINE; RETYPE IT.

An error has occurred in the transmission of information on the telephone line which connects your terminal with the computer system. This information was garbled in transit and the computer was unable to interpret it. This problem may be caused by a terminal malfunction or by a telephone line error. Please type TTY and note the line number given, call (603) 646-3253 to report the problem, then retype the line.

RENAME DENIED

Your RENAME command has not been accepted. This message is always preceded by another error message explaining why the command was not accepted. See section 11 for more information.

REPLACE DENIED

Your REPLACE command has not been accepted. This message is always preceded by another error message explaining why the command was not accepted. See section 11 for more information.

RUN TIME LIMIT EXCEEDED; TYPE 'EXPLAIN RUNTIME'.

Each user number on the Dartmouth Time-Sharing System has associated with it a maximum number of allowable seconds for a single run of a program (see section 5). The program you were running had not terminated when your limit was reached. You may have made a programming error which causes the repeated execution of the same block of statements with the result that the program is in a "loop" from which there is no exit. Check for this error. If your program legitimately requires more than your allowed run time, call (603) 646-2643 or come to Kiewit to make application for an increase.

SAVE DENIED

Your SAVE command has not been accepted. This message is always preceded by another error message explaining why the command was not accepted. See section 11 for more information.

SCRATCH DENIED

Your SCRATCH command has not been accepted. This message is always preceded by another error message explaining why the command was not accepted. See section 11 for more information.

SCRATCH STORAGE QUOTAS EXCEEDED; TYPE 'EXPLAIN SQUOTA'.

You have given a command that requires that your latest alterations be sorted into your current file. The sort procedure cannot be carried out, however, because the sorting requires more scratch storage space than you are allowed to use.

You should work with smaller sections of your file. You may use Dartmouth Edit or the String Editor to segment your file; these systems are discussed in Technical Memoranda 2 and 3 available in room 10, Kiewit basement, (603) 646-2643. If your file is a program, you might reorganize it using subprograms or chaining. As a last alternative, you may apply for a temporary increase in the scratch storage quota on your user number by calling (603) 646-2643 or by coming to Kiewit.

SPEAK!

This is the response to your BUILD or DIRECT command, and you may now begin file building. In particular, you may type in information without line numbers. See the BUILD and DIRECT commands in section 9.

STORAGE QUOTAS EXCEEDED; SCRATCH OR UNSAVE SOME FILES.

The files you have saved in your catalog occupy more storage than your maximum allowed storage. To find out the storage being used, maximum storage allowed, and a list of files in your catalog with the storage used by each file, type CATALOG LENGTH. You must unsave or scratch some files before you retype your command. (If this is absolutely impossible, call (603) 646-2643 to apply for an increase in your storage maximum.) See the CATALOG, UNSAVE, and SCRATCH commands in section nine.

SYSTEM <system name> ABORTED; TYPE 'EXPLAIN JOBABORT'.

Through no fault of yours, a malfunction has occurred in the programming which operates the system you are using. Please give the terminal paper showing your work in the few minutes before you received this message to a Kiewit staff member, or mail it to Kiewit Computation Center, Hanover N.H. 03755. Retry

the command or call (603) 646-3473 or (603) 646-2643 for assistance.

SYSTEM "<what you typed>" NOT FOUND

You have issued a RUN command on a system which does not exist. Check your spelling and retype the comand.

UNSAVE DENIED

Your UNSAVE command has not been accepted. This message is always preceded by another error message explaining why the command was not accepted. See section 11 for more information.

WHAT?

You have typed something other than a legal system command or the line you just typed does not begin with a number. See the list of commands in section 9. You may have typed an l (el) instead of a one as the first character on the line.

WRONG PASSWORD FOR "<file name>"

In an OLD, REPLACE, SCRATCH or UNSAVE command you have specified a password different from the password with which the file was saved. You should check for spelling errors; you may call (603) 646-3474 or (603) 646-2643 for assistance.

YOU ARE NO LONGER CONNECTED TO CONFERENCE "<keyword>"

You were connected to a multiple-terminal conference and have hit a null or break key on your terminal. The master terminal gets the message A USER HAS DROPPED OUT and the conference continues without you.

11. Accesses on Files

When you SAVE a file, access permissions for that file are also stored. These permissions determine what operations may be performed on the file. The permissions and their letter code abbreviations are:

APPEND	A	PUBLIC	P
COMPILE	C	READ	R
FETCH	F	SAVE	S
GROUP	G	WRITE	W
LIST	L	EXECUTE	X

11.1 How to Specify Permissions

You may save a file with a password and may selectively specify access permissions for reference with and without a password. This is done when you SAVE th file. The format of the SAVE command is

```
SAVE ,<password>;<M>;<N>
```

where <password> is any combination of 1 to 8 characters (excluding ",", blank, ":", and ";"). M and N are any combination of the letters RWALSCPGXF. M represents the accesses that are granted when the file is called without a password; N represents the accesses that are granted when a password is supplied. (Do not type the "<" and ">" in your SAVE command.)

The following is a table showing the effect of various forms of the SAVE command. Note the placement of commas and semicolons. When no password is specified, the file is actually given a password of eight blanks.

WHEN YOU TYPE	ACCESSES GRANTED WITHOUT PASSWORD	ACCESSES GRANTED WITH PASSWORD
SAVE	RWALS	RWALS
SAVE ,<password>	none	RWALS
SAVE ;<M>	RWALS + <M>	RWALS + <M>
SAVE ,<password>;<M>	<M>	RWALS + <M>
SAVE ,<password>;<M>;<N>	<M>	RWALS + <N>

11.2 File Operations Associated with Accesses

The access permissions available on a file determine what operations may be performed on that file, whether you are using system commands or accessing the file through a program.

11.2.1 COMPILE, EXECUTE, PUBLIC, GROUP, and FETCH Permissions

COMPILE permission is automatically granted and LIST permission is automatically denied when you SAVE the compiled program resulting from a COMPILE command. In addition, for programs compiled using system BASIC, EXECUTE permission is granted as well. EXECUTE permission is necessary for compiled BASIC programs because of the way these programs are run, and COMPILE permission is needed to run a compiled program written in any programming language.

If you SAVE a file with READ and PUBLIC permissions, it is accessible to any user through an OLD command. READ and GROUP permissions allow access through an OLD command by those users in the same group. (A group is defined by the first 3 letters of the six-character user number.)

If you SAVE a file with FETCH permission (in addition to READ, WRITE, and/or APPEND permissions), that file can be accessed through a program (for example, a BASIC program) being run under any valid DTSS user number. FETCH permission allows the possibility of using a program to read or to modify a file saved in another user's catalog. The accesses the user specified for the file when he saved it determine the operations which may be performed on the file. Therefore, when you save a file with FETCH permission, you must be careful to specify only those access permissions needed for the access capabilities you intended (see section 11.2.3). Type

```
EXPLAIN FETCH
```

for more information.

11.2.2 File Accesses and System Commands

If your current file was created using a NEW command, it is available with RWALS permissions. If your current file was created with an OLD command, permissions available depend on how the file was saved, whether or not the file was passworded, and whether or not the password was specified in the OLD command. (See section 11.1.) In any case, READ (R) permission is always available on your current file. The following is a list of some frequently used system commands and the accesses needed for successful execution of each command.

COMMAND	ACCESSES NEEDED
BACK	R on current file
COMPILE	R on current file

COMMAND	ACCESSES NEEDED
LIST	L on current file
OLD	R if file is saved in your catalog G and R or P and R if saved in another user's catalog
PUNCH	L on current file
RENAME	S on current file
REPLACE	RWA on file to be replaced; S on current file
RUN	R on current file
SAVE	S on current file
SCRATCH	none if SCRATCH; RWA on saved file if SCRATCH <file name>
UNSAVE	RWA on file to be unsaved

11.2.3 Using Files in BASIC Programs

The following is a list of some statements of the programming language BASIC and the permissions needed on the files to which the statements refer for their sucessful execution.

STATEMENT TO BE EXECUTED	ACCESSES NEEDED ON FILE REFERENCED
FILE #	R, W, or A (F is needed, too, if file is saved in another user's catalog)
INPUT #	R
PRINT #	A
READ #	R
SCRATCH #	RWA
WRITE #	A if the file is lengthened; W if some elements in file are destroyed

11.3 Changing Accesses & Adding Passwords

You may want to password a file that is currently saved without a password or to change the accesses on a saved file. Suppose the file in question is named FILEA. The following procedure saves the file with the desired password and permissions. Note that before unsaving FILEA, we save a copy of it under the name FILEB (see SAVE command, section 9). This is a standard and advisable safety precaution to prevent losing the original file. The procedure is as follows:

```
OLD FILEA
SAVE FILEB
UNSAVE
SAVE ,<password>;<M>;<N>
```

where M, N, and <password> are as described in section 11.1. When you have successfully completed the above steps you may unsave the copy of FILEA, which is the file named FILEB, by typing

```
UNSAVE FILEB
```

The above method is the safest way of changing accesses. If you are near your storage limit or FILEA is extremely long, you may get the error message STORAGE QUOTAS EXCEEDED; SCRATCH OR UNSAVE SOME FILES when you try to save a copy of your file. If, in this case, you do not wish to unsave any other files in your catalog, you may still follow the above procedure, omitting the SAVE FILEB and UNSAVE FILEB steps.

11.4 How to Find Accesses on Files

Typing CATALOG ACCESS will result in printing of file name, accesses available without password, and accesses available with password (in that order) for each file in your catalog. To find accesses on particular files in your catalog, type

```
CATALOG ACCESS; <file name> <file name> <file name>
```

as explained in the description of the CATALOG command in section 9.

For example, if you wanted to find the access permissions available without and with a password for files PROGRAM and DATA saved in your catalog, you could type

```
CATALOG ACCESS; PROGRAM DATA
```

12. Glossary of Terms

Acoustic coupler -- a device which allows the user to connect a terminal to the computer system using an ordinary telephone. The coupler is connected to (or built into) the terminal and usually has two round depressions on top into which the telephone receiver fits. The user dials the computer on the telephone, and places the receiver in the coupler; on most couplers, the telephone cord should be at the end of the coupler opposite the cords which connect the coupler to the terminal.

Alter file -- file in which the line-numbered alterations entered from the keyboard are stored. These alterations will be applied to the current file prior to the execution of any command which requires a sort.

Background -- a system on the DTSS which allows a user to execute programs, particularly long-running programs, and use special peripheral devices (for example, high-speed printer, card punch, tape handlers, etc.) without his terminal remaining connected to the computer. See TM008 for details on the BACKGROUND system.

Batch-processing -- a method of computer operation in which programs are submitted for execution and no further opportunity for input is available.

Catalog -- the collection of all a user's saved files.

Compiler -- the system or program which converts the user's program, written in a language such as BASIC or FORTRAN, into a form called machine code which is instructions for direct execution by the computer.

Current file -- the file being created or altered at a given time.

Data Set -- (or "demodulating device") a device which converts signals generated by a terminal into a form which can be transmitted over telephone lines to the computer or vice versa. Data sets usually have a telephone or a telephone dial and receiver associated with them.

DTSS -- Dartmouth Time-Sharing System.

File -- a block of information stored in the computer.

GCOS -- (General Comprehensive Operating System) the batch-processing system available on the Dartmouth G-635 computer.

Hardware -- refers to any piece of machinery in the computer system.

Input/Output Units -- a measure of the amount and frequency of information transferred between various computer equipment while a program is running or while some other activity is being carried out.

Library -- The DTSS public library of files consists of about 450 programs which are considered to be in the public domain and may be accessed and used by anyone with a valid user number.

Object code -- your program after it has been translated into machine language instructions. This is the form your program is in after it has been compiled.

Paper-tape -- a strip of paper with holes punched in it which may be interpreted by the computer terminal as characters or alphanumeric information.

Parity -- Characters are represented and transmitted as a series of six pieces or "bits" of information, each of which may be thought of as either "on" or "off". A seventh piece of information, called the "parity bit" is also transmitted so that the total number of "on" pieces of information is always even (odd). This enables the computer to determine if characters have been transmitted properly since, if any piece of information is garbled or lost, the sum of the "on" pieces will not be the expected even (odd) parity. The computer will sense the discrepancy and print an appropriate error message.

Password -- a series of characters which may be specified in connection with a user number or file name. Only the user should know his own user number password. When a password for a user number or file is typed in, on most terminals it will be printed over by the computer at the terminal so that it is not readable. The use of passwords allows the user protection on the files in his catalog, since only by specifying the correct password may anyone gain access to a user number catalog or to a user's passworded file.

Saved file -- a file saved under a user's catalog so that it may be accessed in subsequent work from a terminal or through a program.

Sort -- as used in this manual, the process of merging the alter file and the current file and arranging the resulting file in increasing line-number order. If two lines have the same line number, the one typed first will be ignored, and any lines without line numbers will be deleted. The resulting file becomes a

the current file, and the alter file contains no information after a sort.

Source program -- the program as you type it into a file using the rules for writing statements in the programming language (such as BASIC) which you are using. This is your program in its uncompiled form. When you type RUN, the program must be compiled or translated into specific instructions which the computer can understand. After the program has been compiled, the computer starts operating on the compiled instructions (or object code) and execution begins.

Terminal -- a communications device which allows the user to send information to and receive information from the computer. Terminals may be of the teletypewriter type (such as the Model , 35, 37, IBM 2741, Friden, etc.) or may be the CRT (Cathode Ray Tube) type with a viewing screen (such as the Datapoint 3000 terminals).

Time-sharing -- a type of computer system that allows several terminals to be connected simultaneously (i.e. "share time") with a single main computer.

User number -- on the DTSS, a six digit number (or a five digit number preceded by a letter) which identifies you to the computer as a valid user.

Supplement to TM022, A User's Guide to the DTSS

Telephone numbers for access to the DTSS (9/72)

The following are telephone numbers to dial to gain access to the Dartmouth Time-Sharing System. If you are on campus and the telephone at your terminal has a green disc on the dial, you can access the computer by dialing a single digit. The appropriate numbers are as follows:

Number	Type of Terminal
7	110 bits per second (Model 33 and 35 Teletypes and Friden terminals)
6	134.5 bits per second (IBM 2741 and Novar terminals)
5	150 bits per second (Model 37 Teletypes)
4	300 bits per second (Terminet 300, and Datapoint 3000 terminals)

From phones with green discs, you may call any other standard campus telephone (except the 4-digit numbers given below) by prefixing the number with a 2.

From other on-campus (i.e. 646 exchange) telephones not equipped for single digit dialing you may access the computer by using the following numbers:

Number	Type of Terminal
5271,5371 5281,5381 5291,5391	110 bits per second
5261	134.5 bits per second
5251	150 bits per second
5241	300 bits per second

If you are off campus, the numbers to dial are as follows (the area code for Hanover is 603):

Number	Type of Terminal
646-5171	110 bits per second
646-5161	134.5 bits per second
646-5151	150 bits per second
646-5141	300 bits per second