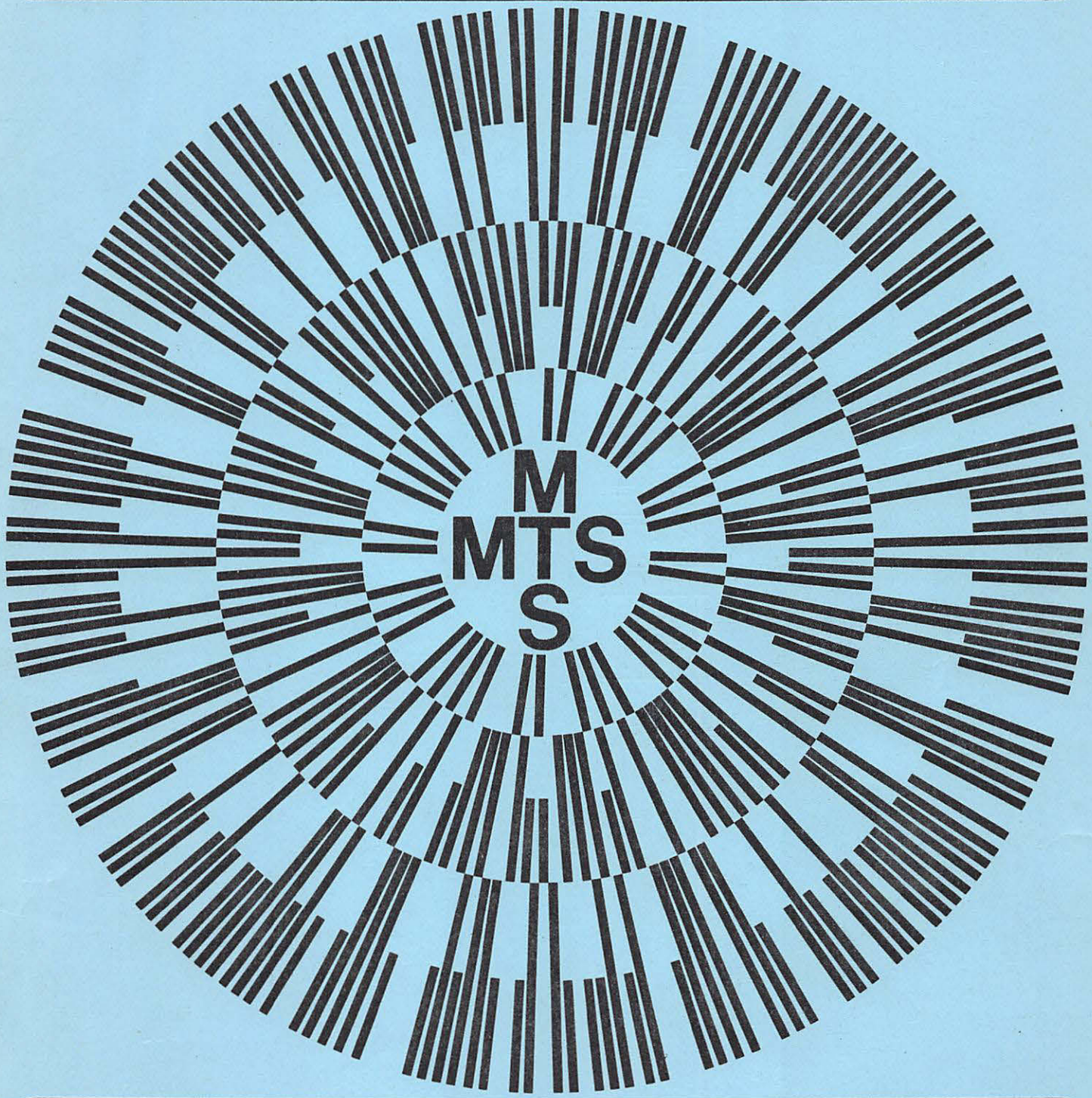
THE UNIVERSITY OF ALBERTA
COMPUTING CENTER
PUBLICATION



INTRODUCTION TO MTS

ACKNOWLEDGEMENTS

THIS MANUAL WAS LARGELY COMPILED FROM MATERIAL PREPARED BY THE STAFF OF THE UNIVERSITY OF MICHIGAN COMPUTING CENTER. THEIR DOCUMENTATION WAS INVALUABLE AND WE ARE INDEBTED TO THEM FOR ALLOWING US TO USE IT. IN PARTICULAR, THE FOLLOWING WERE MOST USEFUL:

MTS USERS' MANUAL, SECOND EDITION, VOLUMES I AND II

MTS USERS' MANUAL, THIRD EDITION, VOLUME 2

INTRODUCTION TO MTS AND THE COMPUTING CENTER (FLANIGAN)

COMPUTING CENTER NEWS ITEMS

COMPUTING CENTER MEMOS

THE COMPUTING CENTER WISHES TO PERSONALLY ACKNOWLEDGE THE ASSISTANCE OF MIKE ALEXANDER AND DON BOETTNER WHO HELPED US TO ESTABLISH MTS AT THE UNIVERSITY OF ALBERTA.

ACKNOWLEDGEMENT SHOULD ALSO BE MADE TO THE COMPUTING CENTRE, UNIVERSITY OF BRITISH COLUMBIA, FOR INFORMATION OBTAINED FROM SOME OF THEIR DOCUMENTATION AND TO I.B.M., WHOSE MANUALS PROVIDED CERTAIN SECTIONS FOR OUR MANUALS.

DISCLAIMER

This MTS manual is a combination of earlier manuals, update notices, memos and limited experience with the system itself. Because of this, certain discrepancies are bound to occur and the Computing Center would appreciate being notified of all differences between what this manual says and what the system actually does.

This publication is intended to represent the current state-of-the-system. However, it should not be construed as an obligation to maintain the system as so stated. The MTS system, like most good systems, is continually being improved. As a result, additions, extentions, changes and deletions will occur. Notice of such changes will be made and provision for a manual updating service has been planned.

Errors, comments and suggestions should be sent to:

Information Coordinator
Computing Center
University of Alberta

INTRODUCTION

The University of Alberta Computing Center is providing a new multi-terminal batch operating system on the IBM 360/67. This system, known as the Michigan Terminal System (MTS), is distributed by the University of Michigan. This manual is an introductory note on MTS and on manuals prepared at the University of Alberta which describe the MTS system. These manuals should be used in conjunction with the IBM Systems Reference Library publications which concern the users' area of interest.

Occasionally changes to various language processors and system facilities and conventions will occur. To communicate these changes to the users as soon as they are viable, a set of news and information files will be maintained. To learn the latest, enter the command:

```
$COPY NEWS:DESCRIBE
```

THE SYSTEM

The core of the MTS system is a set of programs formally called the University of Michigan Multiprogramming System (UMMPS). UMMPS oversees all the computer's multiple-user-oriented activities, including a system called the Michigan Terminal System (MTS). Although MTS is a complex set of programs in itself, to UMMPS it is merely one of a number of jobs. Functionally, we then have a hierarchy of command levels. UMMPS is in complete control of the system; it allocates the use of all hardware. MTS is one of several job programs that run under supervision of UMMPS. The typical user of the system communicates with the system via MTS commands in his job deck. He never sees UMMPS; the interface between MTS and UMMPS is hidden from him. When he makes a request of MTS, MTS asks UMMPS for whatever it needs to handle his request. One of the functions of UMMPS is to intermix batch jobs with time shared jobs initiated at remote terminals. It can assign priority levels to jobs in one or the other mode if there is a significant imbalance in the numbers of jobs awaiting execution.

A user may operate within MTS in one of two modes: conversational mode or batch mode. When operating in conversational mode, the user communicates directly with MTS via a terminal in real time; each command typed by the user at his terminal is immediately processed by MTS, and the results of this processing are then made known to the user at the terminal. The user may then decide what needs to be done and enter the appropriate MTS command. The user thus carries a "conversation" in real time with MTS via his terminal; hence the term "conversational mode" is used to describe this type of interaction.

In batch mode, the user keypunches a card deck which contains all the MTS commands, translator source statements, data lines, and object modules which are needed in order to accomplish the described task (or tasks). The deck is submitted to MTS for processing. It is read into the computer and saved in a file for later processing. The deck and a receipt are returned to the user. The deck is then processed by MTS, together with many other batch decks and eventually the results of this processing (the job output) may be retrieved by the user with his receipt.

The MTS batch function is controlled by the Houston Automatic Spooling Priority System (HASP). HASP, originally developed for the IBM OS system, has been revised to interface with the MTS system and accept MTS jobs to run in batch. (OS jobs cannot be run in the MTS system).

A user communicates with MTS by means of a command language. A typical job deck in batch mode will have command cards, source. program cards, data cards and possibly object program cards. The command cards, placed in precise sequence, obtain the system's attention, tell it what to do, how to do it, and with what data. The command language employed is essentially the same whether the user punches it on a card for batch processing or types it directly on an interactive terminal. Each command to MTS is preceded by a dollar sign ($). For example, every user starts with the command "$SIGNON" and ends with "$SIGNOFF". For convenience these and other commands can be abbreviated. .

Input data and programs may be included in a source deck or they may be stored in the MTS file storage system. Three major types of file exist in the MTS system: public, private permanent, and private temporary.

Public files which usually contain a system program in object form, are available to all users, who can read but not change or delete their contents. Private files are normally available only to the user who has created them, to read from, write into, modify or destroy, as he wishes. Access to them is through use of an I.D. number and password. Temporary private files can be created by a user during the processing of a job. These are automatically destroyed at the next "$SIGNOFF".

All three kinds of files may be organized as line files or sequential files. Line files have a limited size with a very useful indexed sequential structure. Sequential files are useful for storing large amounts of data. Which form a user choses for a private file depends on the nature of the data he wants to store. When a file is created for a user, he may specify the initial amount of storage to be set aside for it. If it is not specified, MTS assigns a small amount of storage. MTS will automatically expand the size later if necessary.

The ease with which a user can duplicate the contents of public files by reading them into his private files is taken for granted by knowledgeable computer users, but non-users might not appreciate the ramifications of this. The "breedability" of source programs, object programs, compilers, problem solutions, sets of raw data, etc. - means, in effect, that each user can have his own private computing universe, one designed to be as large and as complex as he wishes. To the system, however, a file is a file; it handles public and private files in the same way. "Public" or "private" files are more a user-oriented distinction.

Whether a given file is "public" or "private" for a user obviously depends on his relationship to the overall system. File rights and responsibilities diminish with each command interface interposed between UMMPS and, say, an electrical engineering student. The programs a student uses in connection with course problems are public for him but private for his instructor; MTS is public for them but private for some computing center staff members; and the supervisor program of UMMPS is public for all but a very few people. The system is very highly structured; it has to be.

MTS provides a powerful debugging tool for programs written in Assembler 360 (and eventually other) language(s). It is a monitor which provides an environment in which a program is run under strict control. The programmer is allowed to: single step programs, set breakpoints and define procedures to be performed dynamically, display or modify portions of programs, intercept interrupts, go back and rerun indefinitely a portion of code, go back to the start and change the parameter list, write a loader map, etc.

Virtual machine capability (virtual 360's) is currently available under the MTS system. This feature is undergoing continuous development at present.

MTS provides users with main core storage for problem programs in a restricted fashion, i.e., there is a maximum size. The limits imposed have no relation to the physical core size of the computer.

MTS has a comprehensive accounting facility as an integral part of the system. Only authorized persons who have been assigned a signon-id, account number, and project number can use the system. At enrollment time, maximum can be established for disk space, terminal time and plotting time. In addition, an expiration date can be set for a user (useful for students taking a programming course) as well as a maximum dollar charge.

The maximum charge is debited for each batch job or terminal session. Charges for the job are based on such computer resources as CPU time, core storage, connect time (for terminals) or cards read, lines printed, cards punched (for batch). Additional charges are made for disk storage. Users have access to their accounting information at any time. A facility also exists which allows project directors or course instructors to allocate and manipulate funds, disk space, etc., among users within one project.

There are many different kinds of computer languages or translators available to MTS users in the public files: assemblers, compilers, interactive languages, list and string processing languages, simulation languages, and editing and debugging languages.

ASSEMBLERS

*ASMG	System/360 G-level assembler
*PL360	ALGOL syntax assembler for the 360
*STASS360	UBC student interpretive assembler
*1130	IBM 1130 assembler
*1ASR, *8ASR, *9ARS	Assemblers for PDP 1/5/7/8/9 computers

COMPILERS

*FORTG	IBM FORTRAN IV compiler (G-level)
*FORTH	IBM FORTRAN IV COMPILER (H-level)
*PLI	IBM PL/I version 4, level F compiler
*SPL	Student version of PL/I
*SWAT	Small version of WATFOR compiler
*WATFOR	WATFOR compiler
*XPL	Compiler writing language
*MAD	MAD/I, an experimental version of MAD

INTERACTIVE LANGUAGES

*APL	The DOS type III APL/360 interpreter (under development)
*BASIC	The BASIC interpreter
*PIL	Pittsburg Interpretive Language

LIST & STRING PROCESSING LANGUAGES

*LISP	The LISP 1.5 language
*SLIPLIB	A set of subroutines to provide the SLIP language in FORTRAN IV programs via subroutine calls
*SNOBOL4	The SNOBOL4 translator
*UMIST	The U-M Interpretive String language
*L6MAC	A set of macro-definitions to provide the L6 language in assembler programs

SIMULATION LANGUAGES

*GPSS	The IBM General Purpose System Simulation program (GPSS)
*CSMP	The IBM Continuous System Modelling Program (under development)

EDITING & DEBUGGING LANGUAGES

*EDIT	The symbolic file editing language
*DEBUG	The symbolic debugging system

THE MANUALS

Several MTS manuals have been prepared by the University of Alberta Computer Center staff to assist users in getting on the air with MTS. These manuals will be susceptible to updating and change. It is highly recommended that the user maintain his manuals in their latest form. A brief summary of the contents of the manuals is provided here. Additional manuals will be prepared from time to time.

FILES AND DEVICES

A description of the file and device nomenclature used within the MTS system. This manual should be owned and read by all users.

COMMANDS

A detailed definition of each of the MTS commands. It is difficult to even approach the system without knowing these commands so this manual is also a must for MTS users.

EDITOR

This manual describes the file editing processor *EDIT. For conversational users this manual is a must. While *EDIT can be used in batch mode, it is not very convenient - so for batch users it is considered unnecessary.

DEBUG

This manual describes the Symbolic Debugging System monitor, available under MTS, which is only useable in a conversational mode. If you are a conversational user this is very useful, especially if you are familiar with 360 assembler language.

PROGRAM LOADER

This manual describes the University of Michigan program loader which is used indirectly by all users, the card formats accepted by the loader, loader output, several object module handling programs, and the generation and maintenance of private and public library files. Sooner or later most users will have to read this one.

SYSTEM SUBROUTINES

This manual describes the subroutines in the system which are available to users. Many of these routines are quite useful to users.

SUBROUTINE LIBRARIES

Contains a short description of all libraries. It does describe the contents of *LIBRARY in detail and for this reason is worth having.

UTILITIES

This manual describes various utility programs which are provided for users of the MTS systems. Primarily it is a description of the utilities which many users will find helpful.

ACCOUNTING

This manual is of interest only to project directors and class intructors. It may be worth reading but not keeping unless you are administering other users.

VIRTUAL MACHINES

This manual describes ways and means of running a virtual IBM 360 under MTS.

TERMINAL USERS' GUIDE

This manual is for users of IBM 2741 terminals. Teletype devices, should they be supported, will be described in another manual. A useful manual for conversational users of MTS.

BATCH USERS' GUIDE

This manual describes the methods used for MTS batch operations and is recommended for all users since it is extremely unlikely that one could do without MTS batch processing.

MAG TAPE USERS' GUIDE

Users employing magnetic tapes should have a copy of this manual as it outlines tape handling procedures under MTS.

FORGTRAN (G & H)

This manual describes the FORTRAN IV (G & H) operating characteristics under MTS. These processors are the OS/360 versions modified for use within MTS. Detailed language specifications and descriptions will be found in the corresponding IBM publications. A must for every FORTRAN programmer.

PL/I

The IBM PL/I version 4, F level compiler has been adapted to the MTS system. This manual describes restrictions placed upon PL/I by the MTS system and describes several subroutines which are very useful to the PL/I programmer.

ASSEMBLER

This manual describes the Assembler G processor. It also describes the macros provided in the file *SYSMAC and *OSMAC. Assembler programmers will need this manual.

WATFOR

This describes the use of WATFOR and SWAT (a smaller version of WATFOR) under MTS. This is the MTS version of the University of Waterloo FORTRAN compiler.

SNOBOL4

This manual describes the SNOBOL4 processor implemented under MTS.

GLOSSARY

ACCESS TIME: The length of time required to record or retrieve the contents of a particular storage address. For the principal store or fast memory, this is less than one microsecond (a millionth of a second) for any location, because the fast memory allows random access to any item. Access time for items in secondary storage is slower partly because these devices do not allow random access. Retrieving a stored item from a disk file requires a millisecond or more (a thousandth of a second); from a data cell, about 100 milliseconds; from magnetic tape, anywhere from a few milliseconds to minutes - depending upon how many feet of tape must be wound or rewound to reach the desired item.

ADDRESS: A 24-bit binary number associated with a specific memory location. Usually a user will assign a symbol address that system programs convert into a binary coded number which specifies the actual physical location of that address within the computer's memory.

ALGORITHM: An unambiguous, step-by-step, terminating procedure for solving a problem. For computer solution, an algorithm of a problem must be prepared in the form of a program written in a language acceptable to the system.

ASSEMBLER: A program which translates a symbolic machine language into object language. Assembly language is quite like machine language, allowing each 360 instruction to be individually specified.

BATCH MODE, BATCH JOB: A process or task prepared and presented in its entirely, as opposed to an interaction at a remote terminal by a user who issues commands often based on the computer's response to previous commands. This latter mode of interaction is called conversational. Batch-mode jobs are submitted as decks of punched cards which are read into the computer in groups (batches).

BYTE: A collection of eight binary digits; binary digits are called bits.

COMPILER: A program that translates a symbolic, problem-oriented program into an equivalent program in object language.

DATA CELL: A secondary storage device consisting of thin magnetic strips of tape. The data cell retrieves and positions the appropriate strip for reading or writing information from it or onto it.

DISK FILE: A secondary storage device containing one or more plates attached to a rotating spindle. As the plates are rotated, reading and writing heads can pick up or record new information on the thin magnetic film on the surfaces of the plates.

Disk files, run by disk drives, have fairly slow access time, because the file must be rotated into position to gain access to a given storage location; but they have high transfer rates once the information is reached.

FORTRAN: An abbreviation of FORmula TRANslator. A procedure-oriented (symbolic) language similar to MAD, BASIC, and ALGOL. The term refers not only to this language but to source programs written in it and compiling programs for translating them into machine-instruction programs.

HARDWARE: Physical equipment; electro-mechanical devices, including the wiring and cables within and between devices. Software is everything else, including system and user programs in whatever form.

I.D. NUMBER: A four-character identification number assigned MTS users by the Computing Center. To protect against unauthorized use of an I.D. number, it is used in conjunction with a password supplied by the holder of the I.D. number, who may change his password as often as he wishes.

JOB: All of the operations related to processing or execution of a program submitted by a user.

LANGUAGE: A set of symbols for conveying information. There are many different kinds of computer languages or translators available to MTS users in the public files: Assemblers, Compilers, Interactive Languages, List and String Processing Languages, Simulation Languages, and Editing and Debugging Languages.

LIBRARY: A set of programs, and subroutines maintained in secondary storage for use as needed.

LINE FILE: A file consisting of lines - from 1 to 255 characters or bytes in length. Each line is associated with an identifying line number.

MACHINE PROGRAM: A program written in machine language, i.e., binary code.

MACHINE-INSTRUCTION: One of a number of operations the Central Processing Unit is capable of executing. The instruction directs the operations of the CPU. One of the functions of a FORTRAN compiler is to translate symbolic statements into machine-instructions.

MAD: Michigan Algorithmic Decoder, a symbolic language similar to FORTRAN, originally developed by University of Michigan staff for use with the IBM 7090. A different language, MAD/I is under development for use with the System/360 Model 67.

MAGNETIC TAPE DRIVE: The hardware unit for a form of secondary storage in which information is stored on magnetic tape wound on reels that can be mounted and removed from tape drives as necessary. After a tape is mounted, access time for items stored on it varies, depending upon how far it must be wound or rewound to position it for access to a particular location.

OPERATING SYSTEM: The set of system programs designed to organize and control the processing of user programs. Also called the executive system.

OBJECT PROGRAM: A user program after it has been translated. Also called an object deck or object module.

PAGE: A unit of storage equivalent to 4,096 bytes in core, on drum, or disk storage devices. In the System/360 the magnetic-core memory units (boxes) are sub-divided into 128 pages each; the high-speed drum memories are divided into 900 pages each. In disk files each removable disk pack, identified by a volume number, has a capacity of over 7,000 pages.

PASSWORD: Any combination of one to six characters used in conjunction with an I.D. number to gain access to the computer. Its purpose is to prevent unauthorized use of (hence charges against) an I.D. number. The user may change his password as often as he desires (and he is encouraged to change it often), but he can gain access to the computer only by citing the password he last used.

PROGRAM: A procedure for solving a problem by specifying a sequence of commands to be executed by the computer.

REMOTE BATCH MODE: A form of access which has all the characteristics of ordinary batch mode, except that the information in the user's source deck is transmitted over a telephone line to the computer, and the output is returned by phone line for printout at the remote location.

ROUTINE OR SUBROUTINE: A program used to perform a frequently required computational function. Many subroutines, stored in public files on secondary storage, may be called into use by means of commands in the user's program, or may be automatically retrieved and employed by MTS, depending on the function to be performed.

SEQUENTIAL FILE: A file consisting of a set of records, each of which may be of from 1 to 32,767 characters or bytes in length. These records may be read only in a sequential fashion; they are not randomly accessible by record number.

SOFTWARE: All system programs, including language translators, associated with computing; everything but the physical equipment, which is called the hardware. For most users the "system" consists of the hardware and software taken together.

SOURCE PROGRAM: A user program before it is translated. Also called a source deck or source module.

SYMBOLIC PROGRAM: A program written in a symbolic language such as FORTRAN or PL/I.

SYSTEM/360 MODEL 67: The name assigned by the manufacturer, International Business Machines Corporation. The System/360 is a modular arrangement of hardware economical enough for small-scale users but powerful enough for performing extremely sophisticated research. It has the most flexible and complete instruction repertoire of any IBM computer, with its arithmetic operations including decimal arithmetic, fixed-point arithmetic using 32-bit words and 16-bit half-words, and floating-point arithmetic using 32-bit words and 64-bit double-words. The Model 67, designed by IBM in collaboration with Computing Center staff members at the University of Michigan, features hardware to support maximum flexibility and expandability of its time-sharing capacity.

TERMINAL SESSION: A set of interactions with the computer initiated when a user gains access from a remote terminal and terminated when he signs off.

TIME-SHARING: A system in which multiple users are served concurrently, though not simultaneously. The system software - UMMPS and MTS in conjunction - provides the time-sharing capability for the 360/67. In a time-sharing system several programs are executed in a round-robin fashion. The objective is to provide fast response for users at remote terminals and to use the system hardware efficiently.

TRANSLATOR: A program such as the FORTRAN IV or PL/I compiler, which accepts a source program as input and produces an equivalent object program as output. Translators can be classified as compilers (FORTRAN IV or PL/I), assemblers (the 360 assembler language), interpreters (SNOBOL4), and special purpose languages (LISP).