August 13, 1969

TO: Potential Multics Users

FROM: J.H. Saltzer, F.J. Corbató

SUBJECT: Highlights of the Multics System.

This memo gives a brief introduction to those features of Multics which are of interest to potential time-sharing system users.

Multics (from: Multiplexed Information and Computing Service) is the name of a new general purpose computer system under development by the Computer Systems Research group at M. I. T. Project MAC, in cooperation with the General Electric Company and the Bell Telephone Laboratories. This system is designed to be a "computer utility", extending the basic concepts and philosophy of the Compatible Time Sharing System (CTSS, operating now on the IBM 7094 computer) in many directions. Multics is implemented initially on the General Electric 645 computer system, an enhanced relative of the GE 635 computer.

An ability to share data contained within the framework of a general purpose time-sharing system is a unique feature of Multics, and is directly applicable to administrative problems, research requiring a multi-user accessible data base, and general application of the computer to very complicated research problems. The attention paid to mechanisms to provide and control privacy is of direct interest for several of the same applications as well as, for example, medical data. Multics can thus be a valuable tool which provides opportunities for important new research in these areas.

I. THE HARDWARE SYSTEM

The General Electric 645 Computer System is a large-scale, information processing system with most of the features currently found in such systems. If one attempted to classify system, it would fall in the same general category of size as the GE 635, the Univac 1108, and the IBM System 360/65 and 67.

The configuration at M. I. T., shown in figure 1, contains 256,000 36 bit words of core memory (1 μsec. access to 36 bits or 1.3 μsec. access to 72 bits), 1 central processor (1-2 μsec. for most instructions), a high performance paging drum, (moves 1024 words in 4 ms., 16 ms. average latency with hardware eueueing), 34 X 10^6 words of disk storage and a General I/O Controller which handles magnetic tapes, card equipment, and printers, as well as all telecommunications channels. The central processor is built on the GE-635 instruction set, with augmentation to permit control of paging and segmentation hardware.

II AN OVERVIEW OF MULTICS CAPABILITIES

Multics offers a number of capabilities which go well beyond those provided by many other systems. Those which are most significant from the user's point of view are described here. Perhaps the most interesting aspect of all is that a single system encompasses all of these capabilities simultaneously.

1. The ability to be a small user of Multics.

   An important difference between Multics and CTSS is that Multics is expected to provide a really small user with a proportionately small cost. For example, a student can be handed a limited set of tools, can do limited work (perhaps debugging and running small FORTRAN programs,) and expect to receive a bill for resource usage which is substantially smaller than the corresponding CTSS-like user. If all users are small, then of course the number of users can be increased in proportion to their smallness. An underlying consideration throughout the Multics design has been that the simple user should not pay a noticeable extra price for a system which also accommodates the sophisticated user. As an administrative aid, facilities will be provided so that one can solidly restrict any particular user to a specific set of tools, and thereby limit his ability to use up resources.

2. The ability to control sharing.

   There are a variety of applications of a computer system which involve building up a base of information which is to be shared among several individuals. Multics provides facilities in two directions.

   Sharing:

   - CTSS-style links to other users' files
   - Ability to move one's base of operation into another users' file directory (with his permission)
   - Ability to share data in core memory as well as data in files

   Control:

   - Ability to specify precisely to whom, and with what access mode (e.g., read and write permissions are separate and per-user) a piece of data or the entire contents of a subdirectory are available.
   - Ability to revoke access at any time. (A flaw in CTSS on this point has been corrected in the Multics design.

- Ability, using the Multics "protection ring" structure, to force access to a data base to be only via a program supplied by the data base owner. This facility may be used to allow access to aggregate information, such as averages or counts, or specified data entries, without simultaneously giving access to the entire file of raw data, which may be confidential. There are a large number of potential administrative applications of this feature, and as far as is known, Multics is the only general-purpose system which provides it.

3. The virtual memory approach.

   In a direction diametrically opposed to the little user is the person with a difficult research problem requiring a very large addressable memory. The Multics file system, with the aid of a high-performance paging drum, provides this facility in what is often called a "virtual memory" of extent limited only by the totality of secondary storage (drums, disks, etc.) attached to the system. An interesting property of the Multics implementation is that a procedure may be written to operate in a very large virtual memory, but disk, drum, and core resources are used only for those parts of the virtual memory actually touched by the program on that execution. Another very useful property from a programmer's point of view is that files stored in the "file system" are directly accessible to his program by a virtual memory address. This property eliminates the need for explicitly programmed "overlays", "chain links", or "core loads", and also reduces the number of explicitly programmed input and output operations.

4. The Option of Dynamic Linking.

   In constructing a program, or system of programs, it is frequently convenient to begin testing certain features of one program before having written another program which is needed for some cases. Dynamic linking allows the execution of the first program to begin; and a search for the second program is undertaken only when (and if) it is actually called by the first one. This feature also allows a user to freely include in his program a conditional call out to a large and sophisticated error diagnostic program, secure in the knowledge that in all those executions of his program which do not encounter the error, he will not pay the cost of locating, linking, and mapping into his virtual memory the error diagnosis package. It also allows a user borrowing a program to provide a substitute for any subroutine called by that program when he uses it, since he has control over where the system looks to find missing subroutines. In those cases where subroutine "A" calls subroutine "B" every time, there is, of course, no need to use dynamic linking (and the implied library search) between them, so facilities are provided to "bind" "A" and "B" together prior to execution.

5. Configuration flexibility.

An important aspect of the Multics design is that it is actually difficult for a user to write a program which will stop working correctly if the hardware configuration is changed. In response to changing system-wide needs, the amount of core memory, the number of central processors, the amount and nature of secondary storage (disks, drums, etc.), and the type of interactive typewriter consoles may change with time over a range of 2 or 3 to 1 but users do not normally need to change their programs to keep up with the hardware. The system itself can adapt to most major configuration changes (e.g., more memory) by re-initializing itself, an operation which takes a few minutes.

6. The human interface.

Experience with CTSS has proven that ease of use of a time-sharing system is considerably more sensitive to human engineering than is a batch processing system. The Multics command language has for the most part been designed with this aspect in mind. Features such as universal use of a character set with both upper and lower case letters in it and allowing names of files to be 32 characters long are examples of the little things which allow the non-specialist to feel that he does not have todiscover a secret code in order to be an effective user. In a similar vein, a hierarchical file system provides a very useful file organization and bookkeeping aid, so that a user need keep immediately at hand only those files he is working with at the moment. Such a facility is of great assistance when attacking complicated or intricately structured problems.

III LANGUAGES

Multics provides two primary user languages: FORTRAN IV and PL/I. The FORTRAN compiler is fairly standard with a speed of compilation comparable to or a little slower than that of the extremely fast MAD compiler on 7094 CTSS. It is supported by the usual library of math routines and formatted input/output facilities. FORTRAN IV is probably the best language available for low-budget or student use.

The PL/I compiler for Multics is quite interesting, because it offers a very full selection of language facilities, over 300 helpful error diagnostics, and ability to "get at" the advanced features of Multics all at a reasonable cost. On a "seconds to translate a source language page" basis, the PL/I compiler takes about twice as long as does the FORTRAN compiler, and is expected to improve; on the other hand, a page of PL/I program can express considerably more than a page of FORTRAN program. For these reasons, as well as the anticipated wide availability of PL/I on other computer systems, it is the recommended language for subsystem implementers and general research users needing an expressive language.

A few other languages are available or will probably be available. These include:

BASIC - The very popular interpretive language developed at Dartmouth College. It is planned to take the GE-605TSS version of this small system and run it "as is" on the 645.

BCPL - "Basic Compiler Programming Language," developed by Martin Richards, and recommended as a good alternative where machine language might be indicated. (Also available on 7094 CTSS, System 360, and GE-635.)

EPLBSA - A machine language assembler for the GE-645; (not recommended for general use, it is very slow and the machine language is very difficult.

SNOBOL - A string-manipulation language developed by the Bell Telephone Laboratories. (Also available on 7094 CTSS and GE-635.)

QED - A programmable editor which qualifies as a minor interpretive language. (Also available on 7094 CTSS and GE-635.)

TMG - "Transmogrifier" - A compiler-writing tool developed by R. McClure of Texas Instruments.

All of the above languages translate a source program which has been previously stored in a file. Input and editing of source files is done with one of the available text editors, EDM (a close relative of CTSS TYPSET and EDL) or QED. Although interactive, line-by-line syntax checking languages are easily implemented in the Multics environment, none are yet available.

IV RELIABILITY AND PERFORMANCE

An initial version of Multics began operation on a scheduled daily basis for system programming use in September, 1968. It has been scheduled to run on a 24-hour-a-day basis since May 1, 1969. Thus by October 1, 1969, over a year of operational experience will have been obtained. During this time, reliability, functional capabilities, and performance have been brought to the point that, as of June 15, 1969, the system is serving 15 to 20 system programmers simultaneously. This particularly demanding breed of users has found that Multics is a sufficiently useful tool that it has been possible to abandon the previously heavy use of CTSS in favor of Multics.

The configuration shown in Figure 1 is expected to handle about 30 CTSS-class users in October, at a price per user comparable to that of CTSS today. Both smaller and larger users are also runnable on the system in increased and reduced numbers, respectively.

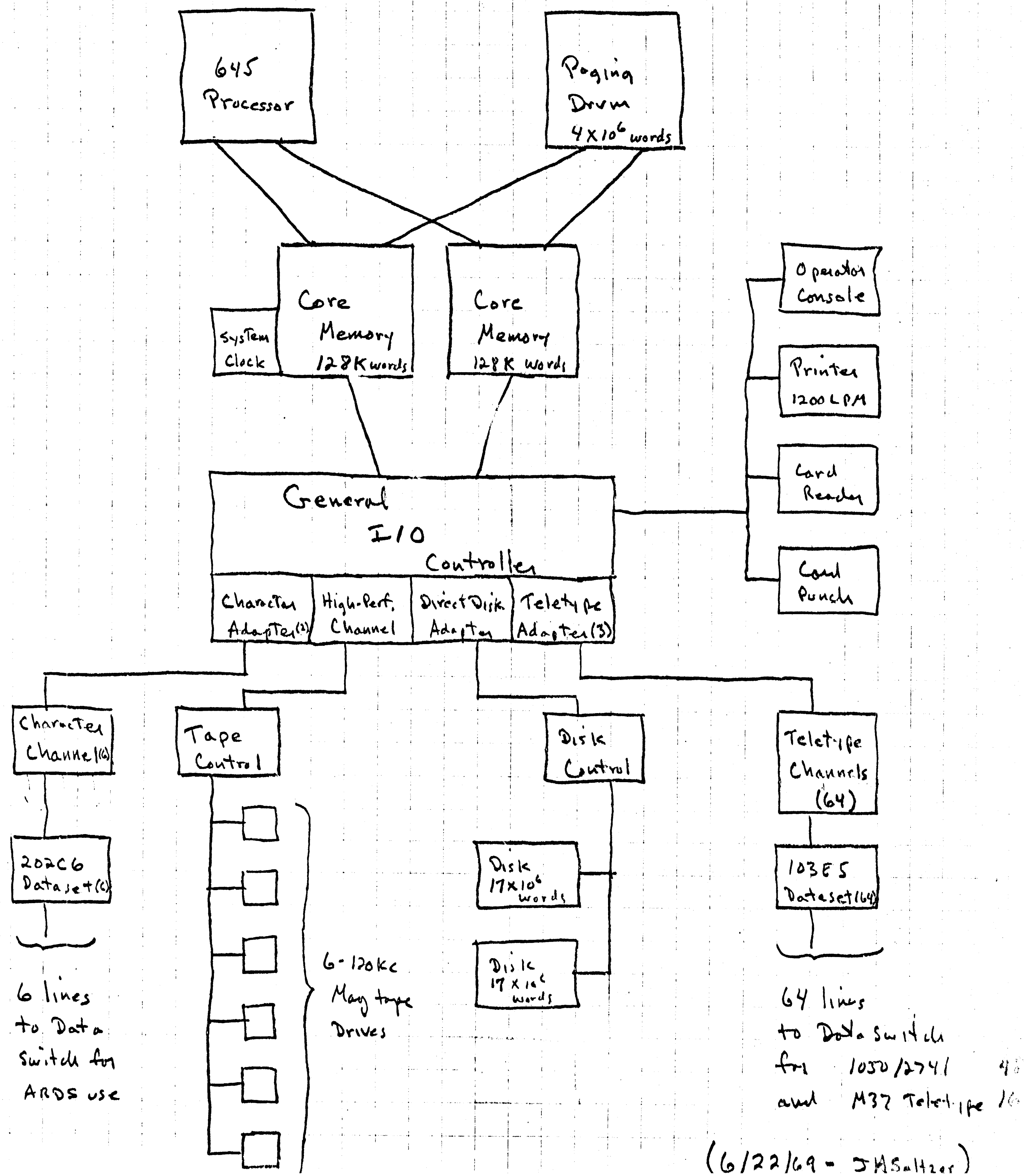
It is expected that as improvements are made to the supervisor and to frequently used commands the cost/performance ratio for the present configuration (especially for smaller users) may improve by as much as a factor of two. Further available hardware improvements are expected to provide as much as another factor of two in cost/performance. Both of these benefits would be realized in terms of either more or bigger users accommodatable on the same hardware configuration.

V THE MULTICS FILE SYSTEM

The Multics file system deserves some special comment.beyond the mention of its sharing/protection features and virtual memory implementation. Following the same style of operation as CTSS, the Multics File System takes on the responsibility for safe keeping of all information placed there by the user. It therefore automatically maintains tape copies of all files which have remained the system for more than an hour. These tapes can be used to reload any user files lost or damaged as a result of hardware or software failures, and may also be used to retrieve individual files damaged by a users own programming blunder.

Each user has an administratively set quota of space which limits the amount of storage he can use, although he may purchase as large an amount of space as he would like; additional disk storage can be added to the 645 in large quantities if necessary.

GE 645 Configuration for Multics Operation



(6/22/69 - JHSaltzer)