INTERDEPARTMENTAL

MASSACHUSETTS INSTITUTE OF TECHNOLOGY *CAMBRIDGE, MASS. 02139*

*from the office of*

June 24, 1969

TO: Information Processing Board

FROM: F. J. Corbató, J. H. Saltzer

SUBJECT: Introducing Multics Service to M. I. T.

This memorandum proposes the introduction of Multics* as a standard computer system of the M. I. T. Information Processing Center.

The proposal begins with a discussion of several of the important capabilities and technical advantages which are offered by the Multics system. They are discussed because it is felt that they are fundamental to meeting of the goal of M. I. T. leadership in sophisticated applications of computer technology to the problems of education, research, administration, and social service. More and more such applications are demanding interactive time-sharing systems operated as computer utilities. Moreover there is great advantage to M. I. T. in having a homogeneous system serve a variety of needs ranging from those of economical student use to those of the elaborate research projects.

An ability to share data contained within the framework of a general purpose time-sharing system is a unique feature of Multics, and directly applicable to administrative problems, social and political science research, 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.

---

* Note: 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.

The second part of this memorandum proposes that the Information Processing Center introduce Multics service at M. I. T. beginning on October 1, 1969. Anticipated usage of the system is discussed in broad terms along with summaries of the proposed configuration, costs, and estimated income.

If Multics service is to be established at M. I. T., it will be necessary to announce this fact to the M. I. T. community. It is felt that much of the material of the present document, especially part one, will serve as a basis for such an announcement. To be properly complete for prospective users, however, further information would have to be included on: 1) the expected prices of the various services and 2) a bibliography of the technical information about the Multics system. Documents on both these topics have already been prepared in draft form.

# I 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.

## 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 MAD 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 wide 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 avilable or will probably be available. These include:

AED - "Algol for Engineering Design," developed by the M. I. T. Electronic Systems Laboratory. (Also available on 7094 CTSS, System 360, and UNIVAC 1108.)

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.)

PAL - "Pedagogical Algorithmic Language" - An interpretive language developed for an E E. Department subject in programming linguistics. (Also available on System 360 and 7094 CTSS.)

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.

## 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.

## II A Proposal to Introduce Multics

The previous sections have explained the technical basis for the introduction of Multics as an offering of the M.I.T. Information Processing Services. The remaining paragraphs discuss several of the issues raised by a proposal to make that offering as of October 1, 1969.

The Information Processing Center should guarantee that the Multics 645 system will have a minimum lifetime at M.I.T. of 3 years. Before that time has elapsed, consideration should be given to taking up the offer of the General Electric Company to provide a program-compatible advanced technology version of the 645 with substantially improved price/performance figures. Since the 645 is a straightforward modification of one of the GE-600 line of computers, continued improvements in that line (e.g., new mass storage, faster core memory, etc.) will in general also be available for Multics upgrading.

The Computer System Research Group of Project MAC will continue indefinitely to provide technical support for the operating system, at the same time as it continues its program of research which includes:

- studying the Multics organization to understand how to provide the same functions with less program complexity
- modifications to Multics to take advantage of new hardware devices as they become available
- exploration into new areas of user facilities made possible by the Multics environment
- incorporation of graphics support packages and connection to the ARPA network

### How Multics Might Be Used

Multics would be available as a standard offering of the M.I T. Information Processing Center. The Multics machine would be viewed as a "production" computer rather than a "software development only" system in M.I.T. Information Processing Center policy. That is, use of the machine to accomplish productive research or educational work (including interactive time-shared use) as well as software development will be encouraged.

Initial discussions with potential users anticipate the following spectrum of use in the first year:

- A substantial number of undergraduate "small users" doing homework assignments, etc., with an intentionally limited set of tools.

- Subsystem developers working out data management systems or advanced programming languages, generally enhancing the useability of Multics by adding to its command repertoire.

- Computer science research on languages and program structures, at Project MAC.

- Departmental and project administrative operations.

- Computer system research group activities (further development and improvements of Multics.)

- Experimental use by other universities with ARPA contracts.

- Miscellaneous former users of CTSS switching to a new long-range base of operations.

- Experimental use by the General Electric Company to explore Multics capabilities as a software producing tool.

The configuration of equipment shown in figure one represents approximately the smallest recommended system configuration. Smaller configurations would have sharply reduced performance for only a slightly lower cost. Larger configurations are, of course, also possible.

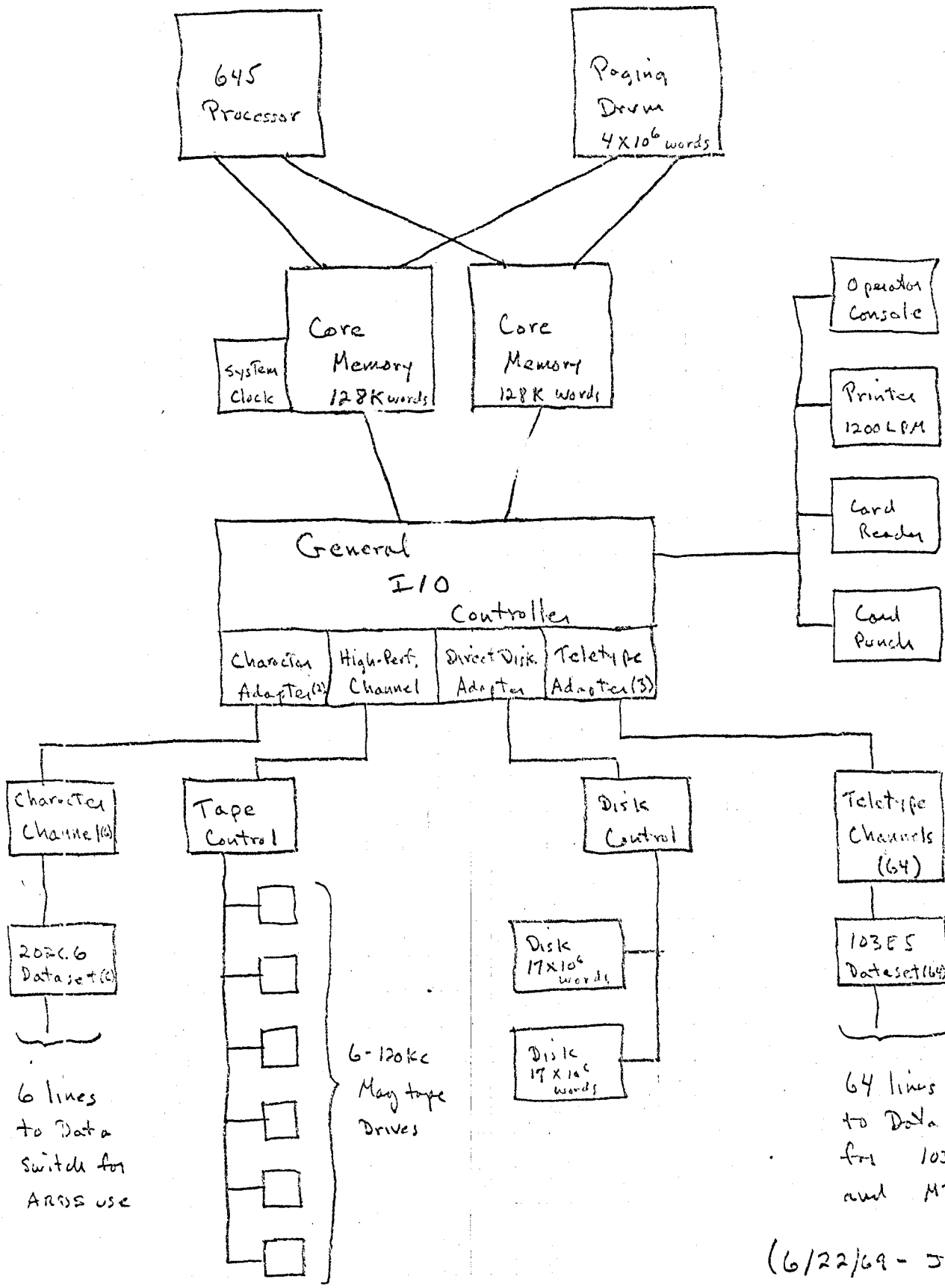
For the nine month period October 1, 1969 to June 30, 1970, the cost of operating this configuration is estimated to be (see appendix 1 for details).

| | | |
|---|---|---|
| GE Equipment rental | $747 | (in thousands of dollars) |
| Second Shift Hardware Maintenance | 38 | |
| Telephone Equipment Rental | 21 | |
| Operation cost | 180 | |
| | $986 | |

A potential spectrum of Multics usage during the nine months is as follows:

| | |
|---|---|
| - Project MAC use, including Multics development | $ 375 |
| - Educational Use | ~~300~~ 200 |
| - Project CAM and other new MAC users | 100 |
| - Other DSR, former CTSS users, outside experimenters, etc. | 325 |
| | $1,000 |

# GE 645 Configuration for Multics Operation




(6/22/69 - JNSaltzer)

APPENDIX

Detailed Breakdown of Equipment Costs for Operating Multics

All costs are stated in dollars per month rental, after taking into consideration educational discounts. Costs are based on prices currently in effect for the equipment.

General Electric equipment rental:

| | | | |
|---|---|---|---|
| 1 | 645 Processor | | $11,870 |
| 2 | Core Memory Controllers, 128 K words each | | 33,300 |
| 1 | Paging Drum, 4 x $10^6$ words | | 9,950 |
| 1 | General I/O Controller | | 5,600 |
| | Tapes: | | |
| 1 | High Performance Channel | $ 544 | |
| 1 | Tape Control | 1,236 | |
| 6 | 120 Kc Mag Tape Drives | 4,300 | 6,080 |
| | Disk: | | |
| 1 | Direct Disk Adapter | 544 | |
| 1 | Disk Control | 2,690 | |
| 2 | DS10 Disks @ 17 x $10^6$ words each | 6,560 | 9,794 |
| | ARDS Terminal Support (for 6 lines): | | |
| 2 | - Character Adapters | 312 | |
| 6 | - Character Channels | 408 | 720 |
| | Typewriter Terminal Support (for 64 lines): | | |
| 3 | - Teletype Adapters | 1,004 | |
| 8 | - 8-line channel and dataset adapters | 1,218 | 2,222 |
| | Card and Printer Equipment | | |
| 1 | - Card reader | 544 | |
| 1 | - Card Punch | 688 | |
| 1 | - Printer 1200ℓpm | 1,333 | 2,565 |
| | Other Equipment | | |
| 1 | - System Clock | 648 | |
| 1 | - Motor Generator Set | 270 | 918 |
| | Total GE Equipment | | $83,019 |
| | Second Shift hardware maintenance @ 5% of rental | | 4,150 |

Telephone Company Rental

| | | |
|---|---|---|
| 64 - 103e Datasets @ $25/mo | 1,600 | |
| 6 - 202c Datasets @ $80/mo | 480 | |
| 2 - Operator's Teletypes @ $120/mo | 240 | $2,320 |

Operations Cost (estimated)

| | |
|---|---|
| (Operators salaries and overhead, paper, cards, supplies, floor space, electricity) | $20,000 |

---

Summary (in thousands of dollars)

| | |
|---|---|
| General Electric Equipment rental | $83.0 |
| Second Shift maintenance | 4.2 |
| Telephone equipment rental | 2.3 |
| Operations cost | 20.0 |
| Total monthly cost | $109.5 |

Notes:

1. Cost of rental of typewriter terminals is presumed to be borne by the user.

2. Some equipment mentioned above can be shared with another 645 computer system if, as at present, two are located together. Presuming a 50-50 split on a monthly rental basis, one has

| | | |
|---|---|---|
| 1 - Disk Controller | - 1,344 | saving |
| 1 - Tape Control | - 618 | saving |
| 1 - Peripheral Switch | + 644 | Additional cost |
| | - 1,318 | net saving on hardware rental |
| Second shift hardware maintenance @ 5% | - 65 | |
| | - 1,383 | saving |

3. A third disk unit of $17 \times 10^6$ words capacity may prove to be needed as user storage needs grow.