March 27, 1969

TO: Information Processing Services Advisory Committee

Dr. Jerome B. Wiesner, Chairman
Professor James D. Bruce
Professor Donald C. Carroll
Mr. William J. Duggan
Professor Aaron Fleisher
Dr. Frederick C. Frick
Professor Robert I. Hulsizer, Jr.
Professor Roy Kaplow
Dr. Myer M. Kessler
Mr. James L. Linderman
Mr. James S. Miller
Mr. Richard G. Mills
Professor Ithiel D. Pool
Professor Frank Press
Professor Daniel Roos
Mr. Robert H. Scott
Mr. John M. Wynne

FROM: F. J. Corbató

The following draft of a memo on Multics Availability at M. I. T., was distributed earlier this year at a meeting of the Committee. Since some members missed that meeting and have asked for copies, it is being redistributed now.

FJC/cpd

INTERDEPARTMENTAL

MASSACHUSETTS INSTITUTE OF TECHNOLOGY *CAMBRIDGE, MASS. 02139*

from the office of

February 12, 1969

TO: Information Processing Services Advisory Committee

FROM: J. H. Saltzer, F. J. Corbató

SUBJECT: Multics Availability Proposal

Enclosed is a draft of a proposed memorandum describing availability of Multics to the M.I.T. community. The memorandum in its present form is primarily intended to invite discussion of the issues surrounding Multics introduction. (The appendix, describing technical aspects of Multics, especially is in need of rewriting before the memorandum is made public.) The memorandum and the announcements it makes have not yet been approved; it should therefore not be reproduced and distributed widely.

Note that the memorandum does not discuss issues of underwriting unused machine resources or of subsidy of interested system enhancers. Both of these issues are important, but should be decoupled from the basic facts about Multics availability. They are being addressed elsewhere.

JHS/mfw

MEMORANDUM

Draft
2/4/69
(by J.H.Saltzer)

TO: M.I.T. Computing Community

FROM: ? { CSR — Saltzer — Mills; MAC — Corbató — etc.; IPC — Licklider — ? }

SUBJECT: Availability of Multics service to the M.I.T. Community.

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. An overview of the technical capabilities and features of Multics in comparison with CTSS appears in appendix I. Further technical information may be obtained from published papers, working documents in the Information Processing Center or Project MAC document rooms, or by direct contact with members of the Computer Systems Research group at Project MAC. A bibliography is included as appendix II.

An Announcement

Work on the development of Multics has progressed to the stage that it is possible to make a formal commitment as to the date of its initial availability to the M.I.T. Community. It is also possible to make reasonable estimates of the cost/performance ratio of the system as compared with CTSS, and of the number of users the present hardware configuration* will be able to support at that time.

---

* 1 central processor, 256,000 36-bit words of core memory, and a high-performance paging drum.

The purpose of this memorandum is to bring these developments to the attention of computer users of M.I.T., particularly to those users of CTSS who are looking for ways of continuing or extending their work, and begin an evaluation of the size of the potential Multics audience at M.I.T.

An initial version of Multics has been operating on a scheduled daily basis for system programming use since September, 1968. Its functional capabilities, reliability, and performance have been steadily improving since that date, and will continue to improve as module simplifications currently in progress are installed in the system. These changes, the last of which will be installed by October 1, 1969, will provide a reliable system with a cost/performance ratio approximately equivalent to that of CTSS today. At that time, the following comparisons should apply:

functional capabilities: substantially more sophisticated than CTSS, but with fewer and more primitive tools to start. Languages will include FORTRAN IV and AED, with PL/I available for experimentation.* A few minor languages such as BCPL, SNOBOL, TMG, and BASIC may be available then or sometime thereafter, but the extensive CTSS repertoire of languages and other tools simply will not exist by then.

reliability: It is expected that Multics will begin public operation with substantially longer running times between system crashes than CTSS did in 1963, but it will initially not be up to the current seasoned CTSS reliability. Long term file storage reliability will be guaranteed, as in CTSS, by automatic magnetic tape backup techniques.

---

* The present PL/I compiler does not offer adequate performance for routine use. Performance of a newer version scheduled for installation by October, while expected to be substantially better, has not yet been verified.

performance: The one-processor 256,000-word configuration will initially handle about 30 CTSS-like users, at a price about the same as for CTSS today.* It is expected that as improvements are made to the supervisor and to frequently used commands the cost/performance ratio for the present configuration may improve by as much as a factor of two; further hardware improvements are expected to provide as much as another factor of two. Both of these benefits would be realized in terms of more (or bigger) users accomodatable on the same hardware configuration. The schedule for these improvements is not firm.

These comparisons describe the system which can be made available for general use on October 1, 1969. In addition, it is expected that experimental use by a small number of users willing to put up with noise and varying daily schedules (and willing to forgo the privilege of complaining about service) will be possible by June 1, 1969.

Reflections on the Announcement

An important pair of questions must be raised by any potential user of Multics:

- Will Multics, or a successor, be available to M.I.T. over a sufficiently long period to make worth the investment implied in choosing it as the basis for my use? (the question of longevity)

- Will it be possible for programs, subsystems, or data banks which I might develop under Multics to be available outside of M.I.T.? (the question of exportability)

The answers to both of these questions are positive, primarily because the Advanced Research Projects Agency (ARPA) of the U.S. Government Department of Defense has insisted that the ingredients for longevity and exportability of Multics must exist.

---

* An important property of Multics is the ability to grow, by adding processor and memory to a single system; within broad limits the cost per user should drop slightly. In principle, the initial system can grow as much as a factor of two or three in this way.

As to longevity, the current 645 hardware is felt to have a minimum lifetime of two to three years before its cost/performance ratio becomes questionable in comparison with other alternatives. In addition, the General Electric Company has committed itself to ARPA to deliver to M.I.T. and any other interested purchasers a successor hardware machine. Preliminary specifications suggest that this successor machine, which has a reasonable delivery schedule, will run Multics with a substantially improved cost/performance competitive with other computing facilities expected to be available anytime soon. Since the Multics hardware consists of a straightforward modification to one of the GE-600 line of computers, there is every reason to believe that continued improvements in that line will continue to be available for Multics upgrading in the same fashion in the future. Thus if Multics becomes heavily used at M.I.T. it will be possible to insure that it will continue to be available for some time.

As to exportability, the following comments apply:

- The three languages AED, FORTRAN IV, and PL/I are widely available at M.I.T. and elsewhere on IBM System/360 and other computers. The job of moving programs and subsystems written in these languages to a different machine is well defined and straightforward in most cases.

- The General Electric Company will, upon request, deliver a 645 system or its successor to any customer with bonafide interest.

- The General Electric Company intends to make a commercial offering of Multics, on a service bureau basis, simultaneous with its introduction at M.I.T. This offering will make Multics, and any developed subsystems which authors may wish to release, available to the public. (The initial market may be restricted to universities and government organizations because of government support of the project.)

Thus we may conclude that all the necessary ingredients for direct exportability also exist. As Multics gains acceptance at M.I.T., one may presume that it will eventually become available elsewhere also. (There are currently two other 645 installations outside of the General Electric Company, at the Bell Telephone Laboratories in Murray Hill, New Jersey and at the Rome Air Development Center, at Rome, New York.)

Taking an overall view, then, the ingredients for longevity and exportability are present. The only thing necessary to insure both is acceptance by the user community.

Finally, a few words need to be said about the nature of Multics availability to M.I.T. Multics would be available through the M.I.T. Information Processing Center. The Multics machine, would be viewed as a "production" computer rather than a "software development only" 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) would be encouraged. This policy contrasts with the alternative M.I.T. Information Processing Center policy of encouraging some machines to be used primarily for sub-system development and chasing developers off as they go into production.

As has been indicated in passing, members of the Computer Systems Research group of M.I.T. Project MAC are prepared to discuss any of the foregoing points, and help answer any questions which might arise.

APPENDIX

COMPARISON OF THE MULTICS SYSTEM WITH CTSS

CTSS and Multics are examples of general purpose, interactive time-sharing computer systems. It is presumed here that the reader is familiar with the capabilities (and perhaps the limitations) of CTSS. Multics is a direct successor to the line of thinking which produced CTSS, with emphasis on the ability to serve a wide range of needs--small jobs or large ones, short computations or long ones, interactive processes or absentee (batch) ones. To this end a number of specific features are worth mentioning. The first seven of these features exist both in CTSS and Multics, but are listed for completeness since they are considered to be of critical importance in both systems.

1. Access Control on Files and File Sharing on a Copy Basis

   Access to files is controlled on both a per-user and per-class-of-user basis. In CTSS while files may be shared, different users must each obtain a copy. Thus in any application where there is frequent updating of the files, the different users will have inconsistent information unless special precautions are taken. Item 12, below, indicates how Multics extends this capability significantly.

2. File Backup and File Retrieval

   File backup refers to the ability to completely restore the secondary storage contents in the event of major catastrophe such as: fire, disk crash etc. File retrieval refers to the ability to restore individual files when they inexplicably become damaged. File retrieval allows an individual to be given some satisfaction without forcing the work of all users of the system to be backed up in time. These two aspects have been key to the success of the CTSS system.

*3. Incremental Creation of File Backup Tapes

   This phrase refers to a refinement of the previous one; it is the ability to continually backup the secondary storage (except for a brief time lag.) The issue becomes an important one as the system

* These features will not be completely implemented in Multics by October, 1969.

size grows and it becomes increasingly awkward to use a brute-force complete dump of secondary storage.

4. Password Logic at Login for Each User

This refers to the ability to authenticate the existence of a user at a terminal, and to properly control thereafter, his access to files within the system. This logic also is the means that "batch" jobs are prevented from having an unauthorized backdoor to the file system since the dispatching of "batch" jobs is done through an on-line console.

5. User Accounting

There is abvious need to be able to account for the diverse use of resources and to distribute the system costs in an equitable fashion. Since Multics is capable of expanding its capacities in various directions this aspect will be expecially important.

6. Automatic Logout and Saving of User Status

Whenever time-sharing system operation must be terminated on either a scheduled or unscheduled basis, it is essential to be able to gracefully shut-down each user's process in such a way that he can restore without error the operation of this process when system operation resumes. This obviously desirable property turns out to be non-trivial to implement, because of the potential for a vastly differently configured system on the resumption of operation.

7. A Command System Which is Open-Ended

This phrase refers to the ability of the user to create programs which are commands of his own choosing and which he can use on the same footing as those supplied with the initial system. This issue has especially large import when it is realized that in an effective time-sharing system many of the better commands are developed by the users rather than by a system programming staff. It is considered important that there be no requirement of reprogramming in order to utilize user contributions.

The remaining features listed are important improvements in Multics which are not present in CTSS.

8. An Expandable Number of Processors Treated Homogeneously

This is a property which allows a system gracefully to expand in capacity and be maintained during 24-hour a day operation. In addition, the property assists in the isolation of major unit malfunctions.

9. An Expandable Amount of Core Memory Treated Homogeneously

The reasons for this feature are similar for those given for processors.

10. The Ability to Quickly Reconfigure the System Without Recompilation

The problem here is that as one begins to develop pools of processors and memory etc., with specific port assignments and cables, the potential number of system configurations rapidly reaches an incredibly large number. For this reason it is essential that the process of initializing a system from a system tape be a generative one since reconfiguration is expected frequently. (Multics initialization currently requires about three minutes to establish the hard core supervisor and about two more minutes to establish the basic system commands.)

11. Isolation of the User from the Configuration

This feature is frequently referred to as having the user program for a virtual machine. It is an essential feature with any machine which changes configuration frequently. CTSS avoided this problem by not being reconfigurable or expandable.

Appendix

12. Sharing of Segments

Access control for files (segments) with the ability to share segments dynamically in core memory is probably the single most important aspect of the Multics system. It is through this seemingly innocent feature that one has the ability to develop systems wherein it is possible for users to work in concert in real-time. Sharing of segments allows cooperating users to selectively merge their computations without losing control of the privacy of their work. Many other properties which are listed below are intimately connected with this idea.

13. Individual Segment Attributes

The ability to have data bases with independently varying lengths, as well as with independent attributes such as read-only, write, etc., gives great flexibility and convenience to a programmer. In particular, he is spared ugly difficulties such as overlays.

14. The Ability to have Rings of Protection Between Sections of a Program

This property, which is invokable by the user, allows him to declare that certain segments are in a separate ring of protection. In this way, he is able to develop a superior/inferior relationship between one group of subroutines and another. This is particularly valuable in applications such as teachers and students interacting by means of a grading program, or in debugging a new subroutine.

15. The Ability to Simultaneously Operate Separate Time-Sharing Systems

This property allows one to have different classes of users who believe they are working with different systems. It also allows a certain amount of system program checkout of future systems without having to take over the machine for another purpose. (Of course, the hardcore functions of the system must be invariant for all users, but these are normally masked by the particular versions of the command language interpreter and login command.) The CTSS "restricted user" is a specific example of an application of this ability.

16. Dynamic Linking or Pre-Binding

Here the user is allowed a choice in Multics of whether to allow subroutines to be discovered on an as-needed basis or to decide to pre-relate subroutines together in a collection so as to eliminate execution-time searching. The property of dynamic linking is felt to be especially important as one develops larger and larger complexes of programs where there are increasingly uncertain paths of the flow of control.

17. More Flexible Library Search Mechanisms

Whether one is loading, binding, or dynamic linking, a directed search is required to discover the location in the file system of a subroutine with a given name. In Multics this search can be through a sequence of directories according to search rules which may vary from user to user. (Initially, this variability is rather limited.)

*18. Interchangeability of Absentee and On-Line Execution of Jobs

What is meant here is the ability to run a job interactively or non-interactively, depending on one's wishes, with a minimum of fuss required to switch the job from one mode to the other. In particular, in Multics there are no restrictions on: the number of such jobs, when one switches, or how often the switching process is reversed. The latter property, like that of automatic logout, is extremely difficult to achieve unless considerable care has been spent in the system design.

19. I/O Streams are Switchable by User Commands

What is meant by this phrase is that it is possible for a user to issue a command such that a program may receive its input from (or send its output to) a different device than previously established, without any modification to the program. It is in this way that much of the simplicity of interchangeability of absentee users and on-line users is achieved. In particular, when an interactive job is to be run non-interactively, the interactive input must be presented to the program as a file of input in place of the typewriter.

20. Hierarchical File System

There is a hierarchical file system with the ability to make links. The hierarchical file system allows the user to refer, with economy of expression, to a large number of different collections of files. Because a hierarchical file system is not always appropriate for the organization of information (for example, all the persons with telephones are not necessarily subordinate to those who have T.V. sets), the extra flexibility of having arbitrary links between directory branches and files in other directories is highly desirable. (The links in the Multics system avoid the design flaw in CTSS which makes removing links difficult to handle.)

*21. Multilevel Storage Management

The drum, disc, and tape memories of Multics are treated by the system as the beginning of a sequence of devices which have successively longer access times. Neither (the bulk of) the system nor the user must concern itself with the device that information is stored on; instead, an automatic algorithm is employed such that the more recently used files are kept on the fast access drum, the intermittently used

files on disc, and the rarely used files on tape. Eventually, it is expected that a user should be able, if he wishes, to override the default action of the algorithm, provided he is prepared to pay the information storage rental costs associated with a fast access device. (Initially, file storage location is not based on activity.)

22. Multiple Processes on One Job

It is possible for any user to develop a program with explicit parallelism by means of separate processes. Either by automatic multi-programming or if the system has multiple processors he can gain an execution speed benefit expressing his problem in this way.

23. Flexibility for New I/O Devices

The functionally modular design of the I/O system is such that it is possible to add conveniently and easily devices as they become of interest to Multics users. The importance of this feature can be understood by observing that CTSS has been successively adapted to 14 different typewriter terminal devices during its brief history.

24. Economic Benefits

a) pure procedure programs
b) multi-programming
c) paging

The above items are considered to be techniques to achieve an end of more effective utilization of the equipment. As such, they should not immediately affect the quality of the services offered any user of the system, but rather the cost. In Multics, at the scale it is presently operating, only minor economics are being realized from these features. But as the scale of a Multics system increases, or as the system becomes used in a specialized application, then these economies will increase. For example, when all users of a system employ a specific sub-system, the pure procedure feature allows an immense reduction in the storage space required per user.

25. ASCII Character Set

The full ASCII 128 character set is uniformly used throughout the system. It is difficult to explain the incredible confusion which arises when there is more than one character set within a given machine. In CTSS, although upper and lower case letters were allowable, it was always necessary to do specialized programming and be extremely conscious of interface problems. A great many specific codes were necessary to constantly convert back and forth between the various conventions. Multics has a clean start and did not inherit the problem of being compatible with batch systems.

In a similar way, the system software supports files and programs with names up to 32 characters in length (CTSS only allows 6 characters). Again this is a feature which is difficult to add later to a system.

26. Recursion and Stacks

Recursion and a stack mechanism for data storage are available as a system convention and are freely useable if the user wishes. Again, like the character set, this feature is difficult to build-in after the fact except on an ad hoc basis. (It should be noted that stack discipline is of great assistance in debugging.)

27. The System is Flexible Enough for Widely Different Applications

For example, one could utilize Multics as a basis for developing an airline reservation system, with a minimum change to the system. Great attention has been paid to such issues as the ability to have as many as 4,000 relatively idle users without either clogging core memory with system data bases or having poor response time for users.

28. There is Full Instruction Decoding in the Hardware

This is a minor point, but it is an improvement over CTSS which had to rely on the sloppy logic of the IBM 7094. The 7094 did something unpredictable whenever it encountered an illegal op-code. The general effect of full decoding is to increase system reliability, minimize inexplicable malfunctions, and increase confidence in protection and privacy hardware.

29. The Ability to have Execute Only Procedures

As time-sharing utilities begin to catch hold, it will become increasingly important for entrepreneurs to have available a variety of means to protect their proprietary investments. One technique is to let users employ programs on a black box basis by allowing the program to be executed, but not read. Multics has this property because of special features in the GE-645 hardware.

30. The System is Designed for Evolution

a) A subset of the PL/I language has been used for implementation.
b) The system is strongly organized along functional lines.
c) System programs are written using the same rules and conventions as those used in programs written by ordinary users.

The above properties have already allowed the Multics system to evolve heavily from its original implementation. While writing a small module, the average system programmer does design iteration in his head, exploring alternatives and rejecting them. In a large system such design iteration is frequently impossible to do before

actually exercising the code. Thus, the initial version of a large system resembles a first draft of a small module. Restructuring and streamlining for simplicity becomes a mandatory requirement for an involved system. It is expected that within a year or two, an overall system design will be determined which meets all of the requirements which have been asked of the Multics system. However, change will still be required for two additional reasons. First, users themselves will begin to stress the system in new directions, directions which are based upon the abilities which they already have. This is a phenomenon which was well observed in the CTSS environment. Secondly, as Multics matures and evolves it will undoubtedly be necessary to anticipate its application to future hardware. New hardware will have different economic tradeoffs for various kinds of components and devices. It is expected that some system restructuring will then become desirable. In any case, it should not be necessary to reflect these changes on to the user, except in the improved cost-performance which he should receive.

This concludes the list of the features which Multics has in contrast to CTSS. The segment sharing feature is considered crucial; a few points like that of op-code decoding are almost trivial. In general, no attempt has been made to estimate here the absolute values of the different features. At present the attitude has been taken that they all are important, although there is a set of priorities as to the order in which starred items will appear.