PROJECT MAC June 14, 1973

Computer Systems Research Division Request for Comments No. 29

CSR DIVISION ANNUAL PROGRESS REPORT, JULY, 1972 -- JUNE, 1973

edited by J. H. Saltzer

The most significant benchmark this year was the announcement by Honeywell Information Systems Inc. that the Multics system, the object of a joint research and development project since 1964, would be offered as a standard product using their 6180 computer system. This announcement heralds completion not only of the Multics project itself, but also of the successful transfer of expertise and knowledge from Project MAC to Honeywell, so that both maintenance and development can continue in Honeywell's hands.

The year was also marked by a continuing, and now essentially complete, transition of the Computer Systems Research Division from a professional programming development team to an academically oriented research organization. Thus, the number of undergraduate and graduate students in the division has climbed from a low of two (in 1966) to 23, and the number of professional programmers has dropped from a high of about 28 (in 1967) to six. Correspondingly, the activities of the division have shifted to research topics which can take advantage of the unique laboratory environment represented by Multics.

These activities fall into four major categories of Computer System Research. The first category is measurement of statistical properties of the presented load on the M.I.T. production Multics site, and development of models of both the presented load and of the system's response to that load. Judging from the number of spontaneous inquiries, both the measurements themselves and the models are of great current interest to manufacturers who seem to be developing product lines with virtual memory and other sophisticated features. The second category of activities are those related to the ARPA network, both working with other network participants in developing protocols, and also in modification to the Multics/ARPAnet interface to respond to new protocols and to better integrate the ARPAnet as a standard facility of a computer utility. The third category of activity is advanced

---

research on the protection of shareable information stored in a multiple-access computer utility. This topic has recently become a hot one, with IBM and several other organizations rushing to obtain some useful results. Since our group has had a long-standing interest in the subject, it is tackling some relatively advanced problems in the area: better definition of the essential central security kernal of a general purpose system, and methods of certifying the correctness of an implementation of that kernal. The fourth and final category of activities are several joint projects with other groups which, as will be explained in detail leter, support the general research goals of the division.

Measurement and Analysis of Computer Systems

Activities in measurement and analysis have been almost exclusively the province of students. The objective in this area is to learn how to predict the performance effect of a proposed system design. On the hypothesis that many future system designs will have functional properties similar to those of Multics, it is an especially interesting system to measure. The availability of a measureable system running with a real load has led to a burst of activity in this area, and the performance of a wide variety of measurements:

- A doctoral thesis developing a hierarchical model of the Multics multiprogramming and demand paging algorithms was completed by Akira Sekino. This thesis was significant for its ability to predict the actual performance of Multics under load, yet using mathematically tractable models. The thesis is available as Project MAC Technical Report TR-103.

- In last year's progress report, a linear model of paging behavior was reported. The model relates the number of memory references (the "headway") between missing pages to the size of the paging memory. For memory sizes below 4 million words, a simple, linear relation was observed. During this year, a paper was written and submitted describing the model, and further measurements have been made exploring the shape of the headway function in the region above 4 million words. These measurements indicate that the linear approximation describes the behavior of the M.I.T. Multics installation quite accurately for memory sizes up to 8 million words, but that an exponential approximation may be better above that point. These measurements are of considerable interest to system designers, who need information about the potential performance effect of the large primary memory systems which are becoming economically feasible with recent advances in Large Scale Integration (LSI) production technology. A Master's thesis by Bernard Greenberg will be available in a Project MAC Technical Report.

- A method of measuring a single user's load on primary memory in a paging environment, in order to estimate program "size" and also to provide a reasonable charge for usage was tried, evaluated, and then added to the standard Multics system. The principle of the method is as follows: for a given size of memory, a "large" program would be expected to cause more missing page faults than a "small" program. Thus, a simple page fault count could provide a crude estimator of program size. In a multiprogramming environment, however, the amount of memory available to a program may be different every time the program runs. Thus a simple page fault count would provide a quite variable estimate. On the other hand, since the page fault count climbs when this memory is smaller, and vice-versa, the

product of memory size and number of page faults should be a relatively stable number, but one which is larger for larger programs. (To the extent that an individual program follows the linear paging model, the measure should be perfectly constant with different memory sizes.) The new charging scheme uses this general strategy, and produces a memory usage measure which seems to be proportional to program size, and which varies with a ten-fold change in memory size by no more than 30 to 50%. Currently, this scheme is documented in the form of three internal working papers, two by Robert Frankston, and one by Prof. J. Saltzer.

- In early 1972, a drum space allocation and access request scheduling strategy called "folding" was implemented on the Multics system. This algorithm traded effective drum storage capacity for drum access time by maintaining multiple identical copies of each page on the drum, spaced equally around the drum circumference. When a page on the drum is to be read, the copy closest to the drum read heads is used, thereby reducing the drum access time.

  The reduction of drum size due to folding causes a redistribution of secondary memory access requests between drum and disk (the third level of memory). An analytic model of the drum behavior under the folding strategy (a variation of a model due to Coffman) was constructed, and using the previously mentioned "linear model" for the paging behavior of the Multics system, an analysis was developed to predict the mean access time of the combined drum-disk memory system as a function of the number of drum folds. Experiments were conducted with several possible configurations of the Multics system under a benchmark load to verify the analysis, and to verify that the number of drum folds actually being used in normal Multics service is optimum. A paper describing these results, by Lee Scheffler, has been submitted to the Fourth ACM Conference on Operating Systems Principles, to be held in October, 1973.

- In the past, disk subsystem design, equipment and configuration selection, and interface algorithm decisions have usually been made informally, without hard data to compare the performances of alternative disk subsystem architectures. A class of infinite-population queueing network models are being developed for predicting the probability density function of disk subsystem access time, given specifications of disk subsystem equipment, configuration, and load of arriving

access requests. The models are unique in that they are effective for the types of loads typically encountered in virtual memory systems which use the disk subsystem as a paging device. The models and analysis methods are applicable to a wide range of disk subsystem types, including both fixed- and movable-head disks, single or multiple channel disk subsystems, and non-pre-emptive priority arrangements for expediting the service of some access requests at the expense of others. Straightforward analytic methods are used to derive relationships between access time, configuration, and load, which are then solved numerically. A set of programs have been developed which evaluate the models and can be used to experiment with proposed disk subsystem configurations. A Master's thesis describing this work, by Lee Scheffler, is in preparation.

. Many problems in computer system performance modelling and evaluation require the manipulation of probability density functions and the solution of complex queueing system problems. Analytic methods are of limited applicability to such problems because unrealistic assumptions often must be made in the name of mathematical tractibility. In the course of research on disk subsystem performance evaluation, a system of programs were developed on Multics for performing numerical computations on probability density functions. Several basic primitives are currently implemented for: creating probability density functions of the common analytic shapes (exponential, hyperexponential, Erlang, normal, uniform, impulse), of any combination of these, or of any shape specified by a table of sample points; for combining probability density functions (weighted summation, convolution); and for displaying, computing statistics (mean, variance, percentile points), and computing derived probability functions (cumulative probability distribution functions). These basic primitives are combined with iterative techniques for the solution of simple G/G/n queueing systems (General arrival discipline/General service discipline/n independent identical servers), and for more complex queueing systems. It is expected that, as performance evaluation research continues, this system of programs prepared by Lee Scheffler will see use in the construction and solution of more accurate models of computer system performance.

- In systems with virtual memory, dynamic control of the level of of multiprogramming is needed to maintain a reasonable balance between unusable idle time and time spent doing page retrieval. For dynamic control, a simple method of estimating the size of each program is needed. A new estimating algorithm, based on extrapolation of the previously observed paging rate of a process, (using the linear paging model for extrapolation) was proposed and implemented, in a test version of Multics. Measurements are not yet complete, but the scheme has already proven to be at least as effective as the currently implemented, very complex, heuristic estimator. There is now growing evidence that the earlier, first attempt at an estimating algorithm treats large programs very poorly. An undergraduate thesis by David Reed has been completed on this subject, and he is continuing to experiment with the technique.

- In a cooperative project with the IBM Cambridge Scientific Center, Michael Chang developed an undergraduate thesis on a methodical simulator of cache-memory designs. This methodology takes into account properties such as instruction overlap and lookahead, and evaluates several performance parameters other than cache memory hit ratio.

In addition to the eight measurement and analysis activities mentioned above, comparison of performance of Multics on the Honeywell 645 computer with that on the newer 6180 computer is underway, but not yet complete. Initial results indicate that the hardware processor is about twice as fast, and that the replacement of the rotating drum with a bulk core has reduced multi-programming and therefore paging by enough to give an overall performance increase factor of three between the two systems. Also, informal measurements of the traffic flowing through the ARPA network attachment have been used to guide the activities of the network group, reported in the next section.

ARPA Network activities

Because of the large amount of production programming which has marked the CSR division's activities in the network area in the past, most work in this area has been carried out by staff programmers rather than the students. This year, student participation is increasing. At the same time, the group is becoming more active in network development activities.

Two significant revisions of the Multics ARPA network software were accomplished during the year. The first of these was to revise an assumption that other hosts have relatively large buffering capabilities. This assumption, made incorrect by wide use of the Terminal Interface Processor, led to a design based on servicing the network once per interaction with a human user or his program at the other site. Widespread use of the TIP, with its small buffers, produced traffic with many network transactions for each message, putting a severe strain on the initial design. The revised design, which responds to small transactions on an interrupt basis, reduces both the real-time delays in using Multics from the network and also the overhead costs at the price of increased complexity in the central core of the supervisor. The revised design has been in operation since October, 1972, and has proven quite satisfactory.

The second major software revision was to convert from a half-duplex network interface to a full duplex one, separating reading and writing onto two hardware channels. This change was made after concluding that the half-duplex connection is not adequately supported by the network itself. (The network resolves certain overload conditions in a manner which drops links to half-duplex connections.)

The full duplex connection required a new hardware interface, which was developed as an undergraduate thesis by Richard Gumpertz with construction help from John Williams, another undergraduate. This new interface was also designed to operate with either a local or a distant network interface port, and to operate with the Honeywell 6180 IOM rather than with the older Honeywell 645 GIOC thereby permitting these two other changes to be anticipated and accepted smoothly. Parts and engineering assistance were supplied by Honeywell, in return for which Honeywell will be permitted to use the design in other attachments of 6000-line computers to ARPA-like networks.

Related to conversion to the Honeywell 6180, which is located in a different building, a second Interface Message Processor (IMP) has been ordered for installation near the 6180. This second IMP will permit attachment of the Honeywell 6180 "development" machine, the planned Project MAC Terminal system, the Artificial Intelligence Laboratory "minirobot", and the M.I.T. 370/165, all in addition to the present three PDP-10's and the 6180 Multics "Service" machine.

In the protocols area, members of the group were quite active in the evolution of the new File Transfer Protocol (used for moving files from one system to another) and the re-design of the Telnet Protocol (used for setting up Teletype-like terminal connections) in conjunction with other members of the Network Working Group. The problem of arranging for file access while maintaining privacy was one primary issue which is still only partly resolved. In another area, Michael Padlipsky has proposed a "unified user-level protocol" which is intended to facilitate use of different operating systems by people who have not made themselves expert in the idiosyncrasies of those systems, by providing a universal interface to common functions.

On the implementation level, the major addition was a File Transfer Protocol server which responds to file transfer requests arriving from other sites. A File Transfer command, and a new Telnet command for use from Multics in accessing other network sites, and an I/O system interface module which permits any Multics program to direct input or output to a network link, are all in experimental use in the user interface area. A first implementation of programs which merge the ARPA network mail facility with the Multics mail facility was completed. A facility to automatically detect the need for and perform typewriter case-mapping was added. This facility (in principle unneeded according to network protocol rules) allows use of Multics from sites which do not yet provide network standard upper/lower case facilities. The re-initialization logic of the Network Control Program has been improved, thus allowing it to automatically respond to and recover from a wide variety of error conditions. The Network driver program has been revised to be compatible with a standard interface to the system operator, and of course, numerous bug fixes were made along the way.

Use of Multics via the ARPA Network has increased over the year. New metering and reporting software was implemented in the Fall, which shows that logins per month increased from 254 to 950 between September, 1972, and April, 1973. At the April level, network use accounts for about 10% of all logins at the M.I.T. Multics site. The metering software has also established that in the same period, the network indirect cost (that is, extra Multics overhead involved because the network connection was used) has dropped from about 100% to about 7%, largely because of the software changes reported above.

Protection of Information

In this category of activity are several long-standing interests as well as a substantial new activity. The long-standing interests relate to providing mechanisms in the Multics design which permit controlled sharing of information with security against unauthorized intrusions.

A doctor's thesis by Michael Schroeder was completed this year, describing a design by which general protected subsystems may be implemented using a domain scheme. A protected subsystem is a collection of programs and data with the property that the data may be accessed only by the programs of the subsystem, and that the programs may be entered only at designated entry points. The general domain model improves on the earlier ring model in that it does not constrain protected systems to be hierarchically arranged when more than one is used in a single computation. Schroeder's thesis goes into details of both a processor architecture and also a file system design which support protected subsystems; both are relatively small (though intricate) departures from typical current-day system designs, and thus appear that they would be quite practical to implement. The thesis is available as Project MAC TR-104.

A second doctor's thesis in the area of protection, by Leo Rotenberg, is in progress. Rotenberg is exploring the consequences of attaching restrictions to information in such a way that even after it is released to a program, the restrictions continue to operate. He has also developed a very interesting method of controlling who may change access specifications in a computer system. Basically, he permits a hierarchical control, but with constraining protocols. With Rotenberg's scheme, for example, one could arrange that a person's manager could have access to his personal files, but only after obtaining the agreement of another, higher-level manager. This example is only one of many possibilities. This thesis will be available as a Project MAC Technical Report when it is completed.

In a related activity, Richard Bratt has completed a bachelor's thesis which involves devising system support software which allows easy construction of user-provided protected subsystems in the protection-ring environment of Multics. Although protection rings are provided by the hardware of the 6180, and two rings are used by the Multics supervisor to protect itself, user applications of protection rings have so far been limited to special cases, since the file system provides no way of cataloguing protected subsystems. Bratt's thesis is concerned with appropriate cataloguing and user interface facilities.

As Honeywell transferred Multics from the 645 to the 6180 computer, the software which simulated rings of protection was dropped out in favor of the 6180 hardware support. Although the processor time to switch rings has dropped dramatically (from 3 milliseconds down to 15 microseconds) the performance improvement so far achieved is modest, since with simulated ring software, ring crossings had been minimized in frequency. The primary gain remains to be realized, as redesign of the central core of the system can now be carried out without the need for minimizing ring-crossings, thereby leading to probable simplication of the central core.

A detailed paper summarizing the design of the Multics information protection system was written by Prof. J. Saltzer. This paper has been submitted to the Fourth ACM Conference on Operating Systems Principles to be held in Yorktown, N.Y., in October, 1973, and it will also be made part of the introduction of the Multics Programmers' Manual.

Work on a new activity has begun: the redesign of the central core of the Multics system (taking advantage of the hardware rings as well as new insight) to produce a potentially auditable version of the parts of the system which affect security. This new activity is a fairly ambitious one, probably requiring about three years, and work this year has been confined to studying various aspects of system organization which can potentially be made more methodical, and therefore simpler, and thus smaller, as needed for auditability. Some of the areas currently being studied include:

- Design (and propagation of the design through the central core of the system) of a more uniform approach to coordination of parallel processes. The basic strategy change is to allow several processes to operate in the same address space. This strategy change will allow, for example, the efficient handling of small network transactions on a scheduled basis rather than an interrupt basis. Many other activities which currently require elaborate coordination strategies (e.g., stopping a process when the user presses his attention key) can similarly be simplified if this change is made. Richard Feiertag is developing this topic.

- Identification of the implementation consequences of using a single, system-wide address space with universal segment identifiers, in contrast with the present Multics scheme which uses a separate address space for each process. A system-wide address space would apparently eliminate large sections of the present supervisor which maintain maps of the individual address spaces; the purpose of this study is to understand just how much simplification could result. Although a revised hardware architecture would be necessary to exploit this simplification completely, even without revised hardware it may be possible to modularize and separate those parts of the system concerned with segment number mapping. Victor Voydock is developing this topic.

- The mechanisms of dynamic linking of programs and data, and of searching through libraries, require an intimate interface with the central supervisor and the mechanisms which initialize a process. The present Multics design avoided the development of this interface by making dynamic linking a protected central supervisor function. A study is underway to figure out how to accomplish the dynamic linking function from outside the supervisor, and thereby remove a large program from the central, protected core of the system. Philippe Janson is developing this topic.

- A doctor's thesis, by David Clark, is exploring a simple but complete I/O architecture which in hardware provides complete separation of independent users, so that the operating system need not include any of the usual, very complex, I/O strategy and interrupt facilities -- they may all operate in the protection environment of the user of the I/O device. This scheme, if implemented, would again provide a substantial reduction in the size and complexity of the central core of an operating system.

As can be seen, all four of the above activities are directed toward simplifying the system so as to make the remaining parts, which implement the security kernal, susceptible to methodical auditing. One final activity in this area has been an attempt to carry out an initial audit of the user/supervisor interface, to see both how many errors would be found and also to learn about

how auditing can be made easier. The general topic of making a system auditable relates closely with several other research and development projects on certification of operating systems currently underway at other sites, and contacts have been set up with these other sites so that work may be coordinated.

Miscellaneous Activities.

Several other activities have been carried out by CSR division members, sometimes in support of other groups with which joint projects are underway.

In a joint project with the Automatic Programming Division, and led by David Reed, a Multics LISP interpreter/compiler system which is completely compatible with the LISP system on the Project MAC PDP-10's was developed. The Multics LISP system was proven operational by the transfer of the Project MAC Symbolic Manipulator (MACSYMA), via the ARPA Net, to Multics, followed by its complete and correct operation. A new LISP manual, describing the language now used on both the PDP-10's and Multics was written by David Moon and Alex Sunguroff.

In a joint project with Honeywell and the M.I.T. Information Processing Center, Jerry Stern undertook as a Master's thesis a redesign of the information backup copying system of Multics. The new design allows scaling up to much larger quantities of on-line storage. The most significant change has been a thorough revision of the file system reloading strategy which should yield a drastic reduction in recovery time. Following a system failure, the extent of file system damage is automatically assessed after which any missing files are automatically recovered from backup tapes. Multics can be made available to users while file recovery is in progress. The design looks sufficiently promising that Honeywell and the I.P.C. have begun to implement it. Stern's thesis will be available as a Project MAC Technical Report.

Project MAC, in a joint venture with the M.I.T. Information Processing Center, has developed a specification for a large (8 million words) primary memory system to be attached to the M.I.T. 6180 (Multics) computer. Rapidly developing technology in Large Scale Integrated MOS circuitry makes such a memory economically practical, and research in Automatic Programming will soon require availability of such a large memory system.

Two undergraduates completed theses, in cooperation with Honeywell. on aspects of the interface between the user and a complex time-sharing system. Daniel Bricklin devised specifications for an interactive program debugger for Multics which is intended to be used at the source language level without knowledge of the underlying implementation. The debugger provides a consistent interface for debugging programs written in several languages and would allow shared pure procedures to be debugged and/or executed simultaneously from several processes without interference. Paul Green designed the code generator for Multics for a new applications programming language developed by Bob Freiberghouse of Honeywell that contains language constructions for dealing with a large, segmented virtual memory and performs detailed run-time checking for semantic errors.

The area of Multics documentation has been the last to be transferred to Honeywell. As a result, the Multics Programmers' Manual, through revision 14, was published by Project MAC, although future revisions are now expected to be handled by Honeywell. As part of revision 12, two new chapters of introduction to the console language and to the programming environment were written. The chapters provide a liberal collection of examples, and greatly ease the problem of a beginner trying to learn the system.

Coordination of planning for the Project MAC terminal system has been carried out with the help of Kenneth Pogran who has also coordinated the installation of data communication cables between the Project MAC building and the Information Processing Center building. The Electronics Systems Laboratory of M.I.T. has agreed to help develop a detailed implementation proposal and a prototype terminal.

Two visitors from Japanese Universities spent six months and one year, respectively, as division members studying the design of Multics. One of these visitors, Prof. Katsuo Ikeda of Kyoto University, has since written a book about the design of Multics, to be published in Japan. A Japanese translation of Prof. E. Organick's book about Multics has been prepared by a previous visitor, Mr. Akio Sasaki, and Toyohiko Kikuchi of the Nippon Electric Company. A final note on Japanese interest in Multics is provided by a letter which declares the intent of the Toshiba Company (a Honeywell licensee) to market Multics in Japan.