M.I.T. LABORATORY FOR COMPUTER SCIENCE
Computer Systems Research Division

October 29, 1976
Request for Comments No. 129

CSR TASK REPORT

by David D. Clark

The attached report covers progress of the Computer Systems Research Division of the Laboratory for Computer Science in the period 1 July to 30 September 1976.

During this quarter, theses were completed by Phil Janson, Masaoki Shibuya, and Art Benjamin.

Dave Reed has designed the basic protocol for the LCS local network.

Doug Wells has completed the major coding of NSW on Multics.

The speed of the Multics-ARPANET connection was increased considerably.

Kernel Design Tasks

I. Multi-Level Traffic Control.

Dave Reed's thesis, which reported the major results of this research, is now available as LCS Technical Report 164. At the beginning of the summer, we decided to proceed with the experimental implementation of the traffic control scheme described in this thesis, with the intention of producing a system which was sufficiently complete to allow us to gather performance information, but which avoided as many extranious issues as possible. The specifications for this implementation are described in RFC 123, recently published. The bottom layer of the two level traffic controller, the layer which supports virtual processors and eliminates the use of interrupts for performing I/O, has been implemented and has run successfully. During the next quarter we propose to produce a version of page control which will run on this level, and then produce a second level traffic controller which reproduces to the necessary extent the interfaces now supplied by PXSS. At this point we will be able to gather sufficient performance information to satisfy ourselves of the viability of the proposed structure.

II. Separating the Functions of Page Control and Segment Control.

Drew Mason has been continuing his attempt to separate the functions of segment control and page control in the virtual memory manager of the Multics supervisor. During this quarter, his Master's thesis proposal was accepted. His current design focuses on three problem areas: how resource control should interact with page control, the types of communication necessary between the layers of the virtual memory manager, and how the functions of the virtual memory manager should be modularized to increase simplicity, understandability, and maintainability. Using the structure of extended type managers, work is almost completed on a model of the virtual memory manager. A partial solution to the "quota problem" has been found. The solution is partial in that it only disentangles resource control from page control and segment control. The full quota problem impacts several other modules in the supervisor. Drew's thesis is scheduled to be completed by January, 1977.

III. Study of System Initialization.

Allen Luniewski spent the summer away from M.I.T., so progress on this project was largely restricted to September, in which period an additional chapter of the thesis was prepared in draft form. At this time, three chapters of his thesis exist, and a fourth is in preparation. This thesis should be complete in the next quarter.

IV. Provision of "Breakproof" Environment for User Programming.

Jeff Goldberg has completed a test implementation of some of the ideas in his thesis. A protected environment was established

for control of the user process, and some major modules in the user domain were protected in this environment, for example event and timer management and I/O for the control environment. Problems related to signalling were uncovered and RFC 113 describing these problems was produced. Several drafts of chapters of this thesis have been written, and he expects to complete his thesis in the next quarter.

V. Restructuring of Page Control.

Bob Mabee continues to make slow progress on the rewriting of Andy Huber's page control, which we are doing in order to separate the intrinsic and extrinsic performance implications of his code. We hope to complete this project shortly.

VI. A Case Study of Intermodule Dependencies in a Virtual Memory Subsystem.

Doug Hunt expects to complete his EE thesis, with the above title, by the end of October, 1976. The purpose of this thesis is to provide some insight into the nature of intermodule dependencies in operating systems. To provide a context for this research, a virtual memory subsystem based on that of Multics as well as some other contemporary subsystems is specified and analyzed. The thesis shows how this case study subsystem can be structured so that a module may manipulate the names of objects provided to it by a second module, and yet not depend upon the correct operation of this second module. The design methodology described in this theses can serve as a guide for eliminating unnecessary intermodule dependencies, at least in subsystems that have characteristics in common with the case study subsystem.

VII. Study of System Reliability and Recovery from Errors.

This project is currently drawing to a close. Harry Forsdick is now producing a document which captures the various insights obtained in the research so far. This document will be completed shortly, and will be distributed as an RFC.

VIII. Organization of the Virtual Memory Mechanism of a Computer System.

The research done under this heading by Phil Janson has been described in detail in the annual report for the period ending 30 June 1976, and also in the previous task report for the period ending March 31, 1976. During this quarter, the thesis was completed and accepted, and is now being printed as an LCS Technical Report.

IX. Input/Output in a Security Kernel.

Gene Ciccarelli has been studying multiplexed terminal I/O in an effort to design a simpler and more unified I/O system. He has studied software for communicating with both the front-end processor and the Arpanet, and several network protocols in order to identify

what properties of these systems are network-dependent, why such dependencies arise, and whether they can be removed from the security kernel, simplifying its design. Modeling the communication system as interconnected buffers elucidates the actions and effects of flow-control, sequencing, error-recovery, and timing-dependencies, which may interact with proceses on either side of the buffers. Certain protocols, however, allow much less interaction and most protocols allow much work to be done outside the kernel. In particular, opening and closing of connections, acknowledging receipt of messages, and large-allocation flow-control can be removed from the kernel.

Network Related Tasks

I. Improvements to Network Mail Facilities.

During the summer, significant work was done on the tools for handling mail on Multics. Steve Swernofsky continued his modification of the net_mail command and the mail queueing facility, adding such features as named mailing lists. He also produced programs necessary for a version of the Multics message segment facility which will allow us to receive large incoming network messages.

This quarter, Charles Davis made considerable progress on the continued development of the read_mail subsystem. A number of new features were introduced, and a number of internal improvements were made. These included a totally new request line parser, making possible an "execute" request that invokes the Multics command processor. Other new features are "save" and "log" requests for storing messages in other mailboxes, and a prototype version of a "reply" request, which is not yet available in the public version of read_mail. Now that classes have started once again progress will be slower, as Charlie resumes his studies. Also during this quarter, an up-to-date version of read_mail was brought up on the Multics system at RADC.

Ken Pogran continues his membership on ARPA's Committee on Computer Aided Human Communication.

Ken Pogran is participating in a group attempting to standardize the format of network mail message headers.

II. Modifications to Multics Answering Service.

The ARPANET file transfer protocol contains a specification for a sequence of commands to perform a login, a sequence that differs from the one normally used to login on Multics. In order to implement file transfer, long ago a program in the Answering Service called the transmogrifier was constructed to translate between the file transfer login and the Multics standard login. This strategy was never satisfactory. Roy Planalp has completed a redesign of the Answering Service so that additional programs can be added to the Answering Service which directly interpret login protocols which differ from the Multics standard. This will enable us to interface directly to file transfer login sequences, as well as other protocols such as RSEXEC. These programs are now scheduled for installation.

III. Development of National Software Works.

During the last quarter, Doug Wells has undertaken a major development effort to get the necessary protocols for NSW implemented and running on Multics. As of the end of September, the major programs required on Multics are written and debugged, and

Doug is now preparing for a partial demonstration of NSW facilities to take place in October.

IV. Development of Maintainance Routines for Network Host Information Database.

There exists on the system a database describing every host accessible through the ARPA network. This database is capable of being modified on line, since changes to the network topology are frequent and sometimes unexpected. The dynamic modification of this database has been, until now, somewhat cumbersome, since no organized set of programs exist to perform the modification. A set of commands that can be used to maintain this database have now been written by Karen Sollins.

V. Measurement and Improvement of Data Rate through the ARPANET.

Ever since the ARPANET connection for Multics was moved from the IMP to the TIP, it has been apparent that the maximum data rate between Multics and the ARPANET have decreased significantly. During the summer, Rick Gumpertz invested significant effort attempting to determine the cause of this slow-down. His primary technique for this investigation was direct examination of the bits flowing between Multics and the ARPANET using an oscilloscope and a very fast strip recorder. In summary, the evidence gathered suggests that the principal bottleneck to the transmission of information is the rate at which the TIP can accept information from Multics, and that this limitation is due to the inability of the TIP to allocate buffers for further incoming packets fast enough. It would appear that the problem is not lack of memory space in the TIP, since the TIP provides the same buffer space for packets as the IMP, but rather a lack of processing power in the TIP. As part of this investigation, we had the interface to Multics on the IMP and TIP modified so that the timing delays introduced were minimized. M.I.T. Service Multics is now connected to the TIP, at port 6, rather than the IMP, at port 44. Measurements performed by Raj Kanodia indicate that the cumulative effect of speeding up the interface, and switching from the TIP to the IMP, is that the maximum data rate for multiple simultaneous transmissions has increased from 26 kilobits per second to 72 kilobits per second, almost a factor of 3 improvement. An RFC will be written at some point describing our observations in more detail.

Research in Distributed Systems

I. LCS Local Network.

This last quarter, Ken Pogran moved much closer to a detailed hardware design for the network. He now intends to use a passive transmission medium similar to the Ethernet, with transceivers based on the design developed by the AI Lab for their Ethernet, and a logic interface modeled after that being developed by Farber for the new ring network to be delivered to ARPA.

II. Development of Protocol for Local Network.

Dave Reed spent the Summer developing a specification for the protocols to be used on our local net. The protocol developed, which he called Data Stream Protocol, or DSP, is an outgrowth of TCP, and differs from TCP only in those respects which relate to our need for simplicity and reliability in our local net. A development describing DSP will be available soon.

III. Backup Using a Data Network.

Art Benjamin has completed his thesis on the topic of using a large, randomly accessible data store connected through a network as a backup device for files stored on a computer.

IV. Recovery from Errors in a Distributed Data Base.

During the summer, Masaoki Shibuya completed a Master's thesis in which he considers the problem of maintaining consistency among duplicate copies of a data base in the face of various sorts of computer and communications failures. He proposed a general strategy for error recovery, based on duplication of update information, which he argued was compatible with various coordination techniques currently proposed for distributed data bases.

V. Input/Output Using Multiple Processes on Multics.

Bob Frankston, Doug Wells, and Gene Ciccarelli have developed a program that provides for the Multics user a character at a time terminal input/output capability. This functionality is achieved using a separate "front-end" process running a stripped-down operating environment. The program currently provides a line-editing facility best suited for high speed display terminals, but also usable on hard-copy terminals.

VI. Evaluation of Direct Information Sharing on Multics.

One of the advertised features of Multics is that two users may directly share a segment, by the simple expedient of referencing it at the same time. It has been suggested, however, that this mechanism is very seldom used in practice. It is very important to

understand if this is true, since direct sharing is a very difficult feature to provide in a distributed system. Warren Montgomery has made some initial explorations of the degree to which sharing is actually exploited on Multics, by analyzing dumps of the SST to determine what segments exist that are writeable and also initiated by more than one process. His preliminary conclusion is that shared writeable segments exist only in very specialized applications in Multics, and that when they do exist, the goal is not unstructured sharing, but rather some highly constrained form of communication, such as message passing through a mailbox. This would imply that the elimination of direct unconstrained sharing between users via segments would not be a hardship to programmers. An RFC describing Warren's results in more detail will be written.

Miscellaneous Tasks

I. Analysis of Pl/1 Programs.

An article entitled "An Analysis of Some Commercial Pl/1 Programs", in a recent issue of the IEEE Transactions on Software Engineering, discussed the general structuring of a variety of Pl/1 programs selected from those written at General Motors. Ross Gale has undertaken a project on Multics to see in what fashion system and user programs here differ from those studied in the article. Our suspicion is that our coding style will differ very strongly from that represented by the GM programs, primarily as a result of the de facto coding standards imposed at the various sites by the relative efficiencies and innefficiencies of the Multics and IBM Pl/1 implementations. We have written the author of the paper and obtained from him listings of his analysis programs, so that we may be sure that the experiments that we perform will exactly duplicate his.