file IPC adm

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Reply to: Project MAC
545 Technology Square
Cambridge, Mass. 02139

Telephone: (617) 864-6900 x6201

May 3, 1971

To: Robert H. Scott

From: Robert C. Goldstein RCG

Subject: Multics Availability

A couple of nights ago at 12:30 AM, forty-five users were logged into Multics. You would probably like to attribute much of this to the spring thesis peak. However, if you check the records, I believe you will find that most of the usage is accounted for by a relatively small number of large projects, such as MacAIMS and the Cambridge Project, whose work, if anything, will expand during the summer. There are currently several bottlenecks in the availability of Multics, some of which, at least, can be attacked without waiting for the follow-on processor to be delivered.

1. System Stability and Reliability

One reason for the heavy night usage has been the poor reliability record of Multics in recent weeks. It could perhaps be argued that the heaviest period of the year is a bad time to experiment with new hardware and software. I am under the impression that the service system currently has priority over the development system with respect to available hardware. This would seem to imply that during the recent very long periods of unavailability, there wasn't even enough working hardware to configure a single system.

2. Additional Memory

It appears that the general problem of system capacity cannot be attacked properly until hardware suitable for running Multics is again available on a production basis. However, in the meantime, one could consider adding additional memory to the system. Because of the relatively slow access characteristics of the high speed drum, it is generally agreed that a larger memory would have a significant effect on system throughput.

3. I/O Capacity

Another problem which is beginning to crop up in a number of places involves increasing the system's capacity for high-speed input/output. I know of no solution to this problem for the short to intermediate term except the possibility of acquiring a terminal IMP and accessing Multics through the ARPA network. I am not sure whether this is feasible, or what it would cost in money and software support. In the longer run, there is a clear need for a more flexible communications controller. I know that Bob Daley was thinking about this problem in the last months before he left, but I have not heard of anything happening since. My impression is that it would take about a year to acquire a Datanet 355, for example, even after the order was placed. It is not too early, therefore, to start worrying about this problem.

4. Scheduling Algorithm

It has also recently become obvious to some of us that the Multics scheduling algorithm does not accurately reflect what we feel your administrative goals ought to be. The current algorithm is heavily biased in favor of small, interactive users, such as those running edm or basic. Not only does the system give preference to these users over those doing more elaborate computations, under heavy load conditions, the system may actually stop serving large users entirely. This has happened to us a number of times recently. It is possible to circumvent this property of the schedule, to some extent, by periodically hitting the interrupt button on the console, thus simulationg an interaction. I am not sure whether or not the amount of system overhead required by this operation is significant. I can tell you that the need to operate in this fashion is not appreciated by the users. When I discussed this recently with Jerry Saltzer, he claimed that the current scheduling algorithm was, in fact, appropriate. It was his view that the system should be run so as to make happy the largest possible number of users, irrespective of the size of their financial commitment. My colleague, Al Strnad, pointed out the essential flaw in this argument. If one were primarily interested in providing good service to users of a text editor and basic, one should not have spent five years developing Multics. The GE Dartmouth system would satisfy this need admirably. Multics was presumably developed to support a somewhat more sophisticated class of user, and it is therefore ridiculous to use a scheduling algorithm which intentionally discriminates against these people.

5. Priority Scheduling

One step which we would very much like to see taken to alleviate this problem would be the addition of a priority mechanism, whereby users requiring a higher level of resource availability, could obtain it by paying a higher rate. The need for this is important; it is not just a question of impatient programmers. The best designed interactive system may become virtually useless if the response is too slow, and the psychological effect of this on the users may persist long after the actual problem has disappeared. I am under the impression that a priority-based scheduler would not be difficult to implement on Multics, and might have a favorable effect on revenue, as well as service.

6. New Hardware

The final issues I would like to raise deal specifically with the follow-on hardware procurement. I am concerned on two accounts. On the one hand, I am afraid that there may be a tendency, in specifying the new configuration, to underestimate the demand for service. This would be particularly unfortunate since many users who are currently operating at a fairly high level of frustration are under the impression that all will be well when the new hardware arrives. My second fear is that the Honeywell people, themselves, will be too conservative in engineering the new hardware, and fail to take advantage of some dramatic recent advantages in technology. I'm not sure whether there is anything we can do about this, but I think it is a very real danger. Even considering when it was built, the GE 645 represented an extremely conservative approach to hardware technology. This clearly increased the cost of the system, but was presumably done in the name of reliability, a goal which does not seem to have been realized. As I look ahead at the future of Multics, I see the need for an ever-larger amount of computing to support each user in addition to a growing user community. This can only be made financially feasible if we ruthlessly exploit all of the available technology. For example, Schweber Electronics recently announced a semiconductor memory with a cycle time of 900 ns and a cost of .7¢ per bit in quantities above 25 million bits. One could certainly consider using such technology to replace the drum in the current system, and I understand that some plans along those lines are in the works. However, since this is a completely random-access memory, one could imagine eliminating the drum entirely, and using this technology to build an extremely large main memory. In this connection also, I would like to remind you of the Computer General laser memory whose specifications I sent to Jerry Grochow several weeks ago.

7. Maintenance Schedule

There is one further step which could be taken immediately which would be of some help. This would involve rescheduling maintenance operations so as to make the system available for users earlier in the morning than is currently the case. Even in the absence of hardware problems, the system rarely seems to be available before 9:30 AM. It appears that a significant demand exists from 8:00 AM onward, and making the system available at this hour might be of some help in reducing the peak load later in the day.

RCG:em
cc: J. C. R. Licklider
Prof. F. J. Corbato
Prof. J. H. Saltzer✓
Dr. D. Yntema
W. J. Burner