MULTICS STAFF BULLETIN

MSB-26

DATE: MARCH 3, 1972

TO: DISTRIBUTION

FROM: R. C. DALEY, J. W. GINTELL

SUBJECT: MULTICS PRIORITY TASK LIST

The purpose of this memo is to establish overall priorities within the Multics development activity throughout the remainder of 1972. These priorities are intended to maximize the effectiveness of the development effort as perceived by the user community and will be used by project management to determine the allocation of the limited personnel and machine resources within the Multics project. Project priorities will be reviewed periodically (e.g., once a month) by the bi-weekly Multics Planning Committee and adjusted as required to meet changing demands.

The tasks presented on the following pages are grouped into the following categories:

1. Critical Priority -- Tasks in this category are felt to be extremely critical in that failure to complete any of the tasks in this category could have serious consequences for the future of the Multics project. These tasks will be allocated primary system access, 1 to 3 hours of grace time and bumping privileges.

2. High Priority -- Tasks in this category are felt to be somewhat less critical in that failure to complete any one of the tasks in this category would probably not be too serious. However, these tasks are felt to be quite important to the future of Multics. High priority

2. (Continued)

tasks will be allocated primary system access, 1/2 to 1 and 1/2 hours grace time and bumping privileges.

3. Medium Priority -- Tasks in this category although non-critical are considered to be highly desirable and are to be given preferential treatment over all other non-critical tasks. Medium priority tasks will be allocated some primary access, 1/2 hour grace time, but no bumping privileges.

4. Low Priority -- This priority is given to all other authorized tasks which do not qualify for a higher priority. Tasks with low priority will be given only secondary access to system resources.

CRITICAL PRIORITY TASKS

The following tasks are categorized as critical priority tasks:

1. Follow-on System Integration -- This task is defined to comprise all tasks necessary to transfer the Multics system and the M.I.T. Multics community to the 645 Follow-on hardware system. This task includes software support of the 645 Follow-on ring hardware as well as minimal support for the extended instruction set (e.g., supported within the pl1_operators segment) but excludes non-critical Follow-on tasks such as the replacement of the GIOC simulation software with a new DATANET-355 software package which exploits the full power of the DN-355.

   The timely and smooth execution of this task holds perhaps the greatest promise for the future of Multics as a stable, reliable and cost effective computing service at M.I.T. and other existing and potential Multics users. It is extremely important that software schedules be met and that the transfer of Multics service to the Follow-on system be executed in a well planned and professional manner.

2. Version II PL/I Checkout -- The checkout and installation of the Version II PL/I compiler is designated as having critical priority in that several major users have critical dependancies on many of the features supported in Version II PL/I. In addition, the Version II PL/I compiler will have a significant effect on system performance in that all PL/I compilations will consume fewer system resources and the improved performance of the resulting object code will have a global effect on all user and system PL/I code.

2. (Continued)

This task comprises all tasks necessary to bring the Version II PL/I compiler and related software to a level approximating the current stability of Version I. At this point there should be no functional reason to prevent any system or user PL/I code from being properly compiled with Version II and executed with correct results. It should also be noted that this task requires the checkout and installation of the file manager with PL/I to support PL/I record I/O.

3. ARPA Network Performance -- It is critically important that Multics performance as perceived through the ARPA network be as good as the performance realized by directly-dialed users. Since this goal has definitely not as yet been reached, this task is established to comprise all tasks necessary to improve the performance of the network software to the point where Multics usage through the network does not consume substantially more resources than direct usage. Machine resources used by this task are supported by Project MAC under the CompNet project which is not directly affected by Multics development priorities. However, this task is to be treated as having critical priority in terms of management attention and personnel resources.

4. Critical Bugs -- Critical priority is also to be assigned to the isolation and repair of critical bugs (i.e., those bugs which cause the system to crash or lose files or otherwise prevent users from accomplishing useful work).

HIGH PRIORITY TASKS

The following tasks are categorized as high priority tasks:

1. Backup Performance Improvement -- The generally poor performance of the Multics backup system constitutes one of the major reasons for Multics non-availability. Not only does the system backup daemon use an excessive amount of machine resources, but its poor performance requires a 2-hour stand-alone save of all disk storage once a day. In addition, if the contents of the Multics hierarchy are lost, the reload procedure may take several hours before the system is ready for general use.

   The goal of the backup performance improvement task is to improve the performance of the backup system to the point where stand-alone disk saves can be eliminated (i.e., when backup can reload the entire hierarchy from backup tapes (incremental and complete) in 3 hours or less).

2. Storage System Enhancements -- This task covers a number of specific enhancements to the Multics on-line storage system intended to improve the performance and maintainability of the storage system as well as introduce some functional enhancements. The following specific subtasks are covered by this task.

   - Improve directory search time.
   - Add performance and lock-contention meters.
   - Introduce more compact branch format for segments with short (i.e., normal) ACL's and for segments under 4096 words in length (i.e., 85% of the current segments in the system).

2. (Continued)

- Support 256K segments. This function is required to support GCOS under Multics and also eliminates the need for multi-segment listing files.

- Enhance and simplify the user access control interface with the introduction of per-segment ring brackets (currently per user), new simplified mode designations (SMA and REW) and initial ACL's(to ultimately replace CACL's).

- Eliminate the need for the User Code Table (UCT) and the maintenance and security problems caused by the UCT.

3. <u>System Control and Accounting</u> -- The following tasks within the system control and accounting area are designated as having high priority.

- Establish project administrator settable limits on the resources (dollars) of some or all users of a project. This feature allows a project administrator to control individual resource utilization within his project.

- Charge interactive resource usage by the shift in which the resources were used rather than by the shift in which the user logged in. This change will lower the resource charges on user-established daemon processes (e.g., the network daemon) which cannot easily log out as the shift changes.

3. (Continued)

- Establish an operator console communication module within system control to allow operator(s) to communicate with daemon and user processes (e.g., to reply to mount requests). This feature will allow much smoother system operation, allow daemon processes to run without private consoles, allow for the introduction of system log files (e.g., for metering, login/logout messages, I/O daemon logs, etc.) as well as support the automated procedures for the mounting and authentication of user tapes (discussed below).

- Establish billing procedures for user tape storage, rental, setup and usage.

4. User Tapes -- This task provides for a series of tasks necessary to establish a straightforward and unencumbered means by which any Multics user willing to spend the money may use Multics tape. The ability of a user to use tape has been one of the most frequently requested enhancements to the system. This task is being undertaken in the following two successive steps:

- The first step (already well underway) is to allow users to mount a tape through the introduction of an interim tape drive allocation strategy and the use of completely manual procedures to control the allocation, mounting and unmounting of tape reels. In this version, the operator is responsible for all tape authentication and must phone the user or use the warn command to type in the user's console if he is unable to mount the requested tape.

4. (Continued)

- The second step will establish automated tape mounting and authentication using the operator console communication facility described above and provide for the automatic charging of tape usage. In addition, the restriction in the tape DCM which prevents the appending of information to a previously written tape will be removed.

5. <u>Performance Analysis</u> -- More effort in the area of performance analysis is required to determine means of reducing the basic cost of an interaction, and to improve the overall response and availability of the system. Personnel will be assigned to this area as specific performance issues are identified.

6. <u>High Priority Bugs</u> -- High priority is also to be assigned to the repair of bugs which although non-critical are judged to be extremely troublesome to the user community.

MEDIUM PRIORITY TASKS

The following tasks are categorized as medium priority tasks:

1. Standard Object Segment -- This task provides for the establishment of the new standard object segment format to eliminate the need for maintaining the multiple variations now supported. The new standard is much broader and more precisely defined and once supported by the system and documented in the MPM it should eliminate a great deal of the current confusion on this subject for all system programmers and users who must deal directly with object segments.

   This task is to be implemented by upgrading all programs which manipulate and produce object segments to adhere to the standard (e.g., PL/I, Fortran, alm, binder, linker, etc.) and publishing the new standard in the MPM.

2. Recompilation of System in Version II PL/I -- This task provides for the recompilation of the entire Multics system in version II PL/I to reduce the overall size and generally improve the performance of the system. Once the Version II compiler is deemed ready for system use, personnel will be assigned to speed the conversion of the system from Version I to Version II PL/I. In addition, it is expected that the usual techniques to force conversion of any modules modified and reinstalled after a specified date will also be applied.

3. Upgrade of System Installation Tools -- This task provides for a substantial upgrade in the procedures and commands used for system installation and record keeping to eliminate a number of deficiencies and restrictions in the current installation tools.

4. Ring Support for User Subsystem -- A small amount of work is required to allow users to place subsystem privileged primitives in ring 3 with the restriction that subsystem modules cannot share I/O attachments, event channels or condition handlers with the outer ring. For the interim, subsystem access will be restricted to within a pre-specified project (i.e., if A wants to use B's subsystem, he must login to B's project and vice versa). In the long run, the solution proposed by Schroeder could remove this restriction. However, user rings implemented with the above restrictions will suffice to meet the immediate needs of our users.

5. I/O System Upgrade -- This task consists of the merging of the concepts of the Multics file manager (e.g., logical records, record keys, record locks, and access methods) with concepts of the Multics I/O system to form what is commonly referred to as a Data Management System. The resulting merger will increase the functional capability of both the file manager and the I/O system, eliminate semantic and functional incompatibilities, allow greater compatibility for data interchange with other systems (e.g., OS/360) and allow users of higher level languages to use the I/O system within the semantics of the higher level language.

5. (Continued)

This task can be accomplished in a totally upward-compatible manner in the following steps:

- Extend the I/O switch to include new calls to deal with concepts such as logical records, record keys, record locks and access methods.

- Define current I/O system read/write opration as the Linear Access Method (LAM) to fit with the new semantics without changing anything else.

- Integrate the file manager with the I/O switch.

- Establish new standards for DIM's as to which calls are to be supported by all DIM's.

6. LISP Improvements -- Some enhancements and extensions to Multics LISP are required to incorporate it as an effective tool in the new Project MAC APS effort. It is expected that Project MAC will fund the machine resources required for this task as for the CompNet project. However, this task is to be treated as having medium priority in terms of management attention and personnel resources.

7. Bug Fixing and Improvements -- The fixing of bugs not qualifying for critical or high priority may be granted medium or low priority at the discretion of project management. In addition, certain system improvements such as the removal of a troublesome restriction may also qualify for medium priority.

LOW PRIORITY TASKS

The following tasks are categorized as low priority tasks:

1. Non-Critical Follow-on Tasks -- The following tasks associated with the Follow-on integration activity are currently designated as low priority but will certainly be raised in priority as the more critical tasks are completed.

   - The PL/I compiler is to be modified to support the extended instruction set with in-line code compiled to the user's object program.

   - The DN-355 software is to be redesigned and rewritten to fully exploit the capabilities of the DN-355 (e.g., inform dialed-in users if system is down, handle line discipline, etc.).

2. APL Performance Improvement -- This task encompasses all tasks associated with the development, improvement and installation of the APL interpreter.

3. Record I/O Support for Tape -- This task provides for the support of the Sequential Access Method (SAM) for records recorded on Multics standard tape within the Multics data management system. The result will allow PL/I programmers to access magnetic tape files with standard PL/I record I/O.

4. On-line Documentation -- This task consists of a general upgrade of Multics on-line documentation. Specifically, a number of info (help) segments must be upgraded and consideration is to be given to the installation and maintenance of the peruse_text facility.

5. Bug Fixing and Improvements -- Bugs and improvements which cannot qualify for higher priority will be assigned low priority. Improvements currently assigned to this category include the seek-overlap disk DIM, core metering and user modification of his password.

System Priority for MPM and SysLib -- Since system use by the MPM project counts as only .5 load units per user and due to the special requirements of this project, primary access will be allocated to the MPM project.

The SysLib activity however, derives its priority from the priority of the task requesting the system installation. In addition, a system installation should not be bumped once it has begun. These facts require that system installations must have primary access to the system but that this access should be used with discretion by the installation group and in accordance with the priority of the task requesting the installation.

Documentation -- A strong effort must be made to upgrade and complete the Multics system design documentation during the remainder of 1972, to allow the Multics design to be exported within HIS, PDO and other potential Multics sites. A continuing effort is also required to continue to upgrade our user documentation. Documentation does not compete for machine resources (outside the MPM project) but is considered to be high priority in all other respects.

OTHER TASK AREAS

A number of other tasks and potential tasks have been identified but for various reasons could not be assigned a priority within the same framework as the tasks listed above.

1. Backup System Redesign -- It is clear that the current backup system is going to prove inadequate to support a high volume of on-line storage. In addition, the somewhat related areas of detachable hierarchy (detachable packs) and off-line segments are in need of some strong design effort.

2. On-line T&D -- A great deal more attention should be given to the subject of on-line T&D programs to improve system availability and allow certain hardware bugs to be tracked down while the system is running and available to users.

3. New IPC -- The IPC facility is believed to be in need of redesign for improved performance and enhanced function. Rich Feiertag has several ideas on the subject which might be turned into a design and implementation proposal.

4. System Initialization -- The recoding of system initialization to pre-link the system at MST generation time and to pack system modules more tightly has been proposed several times. The issue appears worthy of investigation at the least, perhaps by a student.

5. Graphic and Display Terminal Support -- More work is required in the area of support for high performance graphic and display terminals. In this area, a good deal of effort is required to determine precisely what the requirements are and to make a proposal.

6. Full Ring Support for User Subsystems -- A proposal for the "full implementation" of rings has been made by M. D. Schroeder. This proposal should be evaluated in terms of design, performance, cost, and user requirements and a determination should be made as to what action if any is to be taken.

7. New Basic -- A new Basic compiler is in use at Dartmouth as the standard version and the old version (the one we use) is to be deleted. The adaptation of the new Basic might make a good student project if we are interested.

8. Dynamo -- A "90% complete" version of Dynamo for Multics has been written in PL/I if anyone is picking up the pieces.

9. COBOL -- Work is scheduled to start on a Multics COBOL compiler fairly soon. COBOL, although of little interest at M.I.T., is a clear requirement for any system to compete as a commercial product.

10. RJP Daemon -- It is highly desirable for Multics users to be able to submit OS/360 batch jobs over a BISYNC communication link to the batch facility and to receive files in return. This link would allow greater interchange between M.I.T. batch and Multics users and greatly simplify communication between the two systems.