To: Distribution

From: Steve Webber

Subject: Performance Group Task Report

Date: 6/24/74

General

The performance group has been spending much of its recent time studying several new proposals for changes to the system that are of a nontrivial nature. Writeups for many of these proposals will be coming out soon and it is hoped that some of the more promising proposals may be agreed upon soon and development begin in the next month or so.

Studies have also begun into a detailed measurement of the system in order to gain insight into the excesive paging the system exhibits. The preliminary studies show little that is new, but do point out the large paging load caused by highly used system functions such as the Storage System, the basic command loop, the TTY dim and the linker. Separate studies into minimizing each of these separately will be undertaken as parts of projects such as combining the linkage, lot and stack into a single segment.

Page Control and Traffic Control Changes

Some of the work done recently has been modeling the paging behavior of the system in relation to load, reference patterns and possible interactions with the scheduler. Several interesting discoveries have been made and a unified proposal describing a combination of several recently discussed changes to page control is being worked on. The new proposal combines the concepts of page pooling, page stealing and process swapping. Some of the benefits that may result from the new scheme are 1) better utilization of resources, 2) more equitable core accounting, 3) more uniform thoughput as the load increases and 4) the capability of giving special consideration to installation selected subsystems in order to make their use less costly.

The other important study undertaken has been the development of a new traffic controller scheme which allows for better administrative control over the balance of the system as well as more uniform response times for interactive requests as load increases. A preliminary version of a scheduler was described in MTB-066, and this proposal has been extended to

---

include an interface to page control which is general enough to allow process swapping -- a descendent of the prepage mechanism used on the 645. An interesting feature of the scheduler is that core use limits will be governed not only by mean time between page faults but also a measure of perprocess thrashing (as well as the working set estimates). A detailed proposal of the new scheduler will come out soon after the unified page control proposal.

Some minor, yet important, changes are also planned to the bulk store DIM. These changes will allow for the software and possible hardware queueing of requests thereby making it less likely that the page fault handler will have to loop waiting for bulk store i/o to complete. All write requests will be deferred until after the read for a fault is issued. While the read is taking place, the software will queue the writes (if they exist) thereby utilizing more fully the CPU. The result of this change may mean as much as a 10% to 15% decrease in the page fault handling time. This change will be even more beneficial on multiple CPU configurations.

Miscellaneous Tasks

There are several tasks being investigated which will have an effect on system performance. A new call/push/return sequence has been tested which is 20% faster than the current scheme and appears to be fully compatible, although it can not be known what tricks and assumptions users have coded into their subsystems. The new scheme removes unnecessary code and defers until later the determination of the exact entry point values. A further simplification is the addition of a new hardware feature in the RTCD instruction to make it work as the 645 did (and restore indicators).

The projects to speed up the TTY DIM and the tape interfaces are being described in separate MTRs.

Directory Control and Segment Control Speedups

Several changes to directory and segment control are being made as well as several other proposals considered. The programs that exist now are all being converted to good EIS code and the algorithms are being cleaned up and made more efficient. The entire locking strategy is being reviewed in light of some newly found deficiencies. If some simplifications can be made here it could have a considerable effect. A study is also being made into a new set of storage system primitives which could have a considerable performance improvement as the system is converted to use the new primitives. In particular, moving the function of "expand_path_" into ring 0, although not pleasing from a "kernel audit" point of view, makes the mapping from relative pathname to segment pointer much more efficient than it is today.

Another change being proposed is the restructuring of the Known Segment Table (KST). The new structure would contain several more items for each known segment including effective mode, bitcount and extended access (for segments and directories). This simple change (a slight extension of the way things used to be before 1968) makes access checking much more efficient both in execution time and in paging overhead. The more important directory access (SMA) will make directory page references much less common. From our studies, such references have been the cause of a great deal of paging. The reformatted KST will also be compatible with the prelinking proposal discussed below.

Prelinking Project

The prelinking proposal has been tabled for a while so that some of the ideas contained therein could settle down. The entire proposal is now being documented and although prelinking is a fairly large task that will include such changes as a modification to the compilers, it is still being considered earnestly as a potential performance improvement that should not be overlooked. Due to the magnitude of the job, however, the implementation (if at all) will probably be delayed until after some of the simpler improvements have been made.

Cache Software

The cache software has been completely checked out with a noncached 2-CPU configuration and is currently being checked out with CPU's with cache in Phoenix. It is not expected to cause any trouble and the cache software will be installed in the standard system as soon as possible. The cache system (hardware and software) runs about 20% (conservative) faster and is a prime factor in our estimated performance improvements.

New Backup Primitives

The design and checkout of the two new backup primitives is proceeding well, on schedule. The integration of the new primitives will begin soon. The expected benefit of these primitives will have a considerable effect at MIT where extensive resources are devoted to backup.

Command Environment Changes

There are several small tasks being undertaken to optimize the command loop execution as well as a large study project to try to integrate all of the important programs of the command environment into an efficient, consistent package. The new command language proposed in MTB-063 is part of this last study. Hopefully, a new design can be agreed upon. In the interim,

several critical commands are being converted to good EIS code and others are being reevaluated. The project of combining the linkage, stack and lot into a single segment will have a direct bearing on the working sets of the command environment as well as to make better use of the AST pools. This mechanism was made workable on the 645 and will soon be redone for the 6180. The system has changed so much since the original implementation that the project will be redone from scratch.

GROUP Multics Development Group DATE 06/26/74 PAGE 1 / 6

PROJECT Performance Improvement Project AREA Performance Improvement Tasks

TASK DESCRIPTION	PERSONNEL	START	FINISH	M-W	CHANGES-STATUS
Page Control/Traffic Control Changes					
Write and debug page control -- traffic control modeling program.	Webber Mullen	03/01/74	05/15/74	6	On-going task as new features are tried.
Perform modeling experiments with updated models.	Webber Mullen	03/15/74		12	On-going task.
Write up page pool proposal.	Webber	06/10/74	06/17/74	1	To be integrated with unified writeup.
Write up new paging removal algorithm proposals.	Webber	06/10/74	06/24/74	2	To be integrated with unified writeup.
Write new integrated Bulk Store Dim.	Snyder	07/01/74	07/15/74	2	
Debug new Bulk Store Dim.	Snyder	07/15/74	07/29/74	2	
Model new scheduler proposal.	Webber Mullen	07/15/74	08/15/74	4	On-going task as new features and extensions are tried.
Write proposal for integrated scheduler and page control changes.	Webber			3	Cannot start until more modeling and experimentation is complete.

TASK DESCRIPTION	PERSONNEL	START	FINISH	M-W	CHANGES-STATUS
Miscellaneous Tasks					
Study and develop faster call/push/return sequences	Webber	05/20/74	07/01/74	4	Preliminary version is 20% faster for no args case
Study and document proposal to swap process images at post-purge and pre page time.	Webber	06/01/74	07/15/74	4	
TTY Dim Speedups					See separate MTR.
TAPE Dim Speedups					See separate MTRs.

GROUP Multics Development Group DATE 06/27/74 PAGE 3 / 6

PROJECT Performance Improvement Project AREA Performance Improvement Tasks

TASK DESCRIPTION	PERSONNEL	START	FINISH	M-W	CHANGES-STATUS
Directory Control/Segment Control Speedups					
Study directory control interfaces for efficiency and uniformity	Stone Webber Staff				Large job that has user impact if done correctly
Review locking code and strategies to improve execution time, etc.	Stone Greenberg Webber	07/15/74	08/15/74	4	
Convert directory control and segment control to EIS PL/I.	Stone	01/01/74	06/15/74	8	A few programs left.
Recode find_ for efficiency (EIS), readability and structuring.	Stone	06/15/74	07/15/74	4	
Command Environment Speedups					
Combine stack, combined linkage and LOT into single segment				4	Done once. Previous work must be found and upgraded.
Convert commands to EIS PL/I.	Staff				Continuing task until complete. (Several months, some important ones are done.)
Integrate the programs of the command environment to minimize working set.	Barr				May be part of new command language implementation.

PROJECT Performance Improvement Project AREA Performance Improvement Tasks

TASK DESCRIPTION	PERSONNEL	START	FINISH	M-W	CHANGES-STATUS
Prelinking Project					
Document proposal for prelinking the system at bootload time.	Webber	06/15/74	07/01/74	2	
Document tentative findings from preliminary prelinking study.	Webber	06/15/74	06/22/74	1	
Design and document proposed changes to the format of the KST to make access checking and prelinking efficient.	Webber Stone	07/01/74	07/15/74	1	
Cache Software Development					
Implement Cache software for 1 CPU system.	Greenberg	02/25/74	04/01/74	5	DONE
Implement and checkout 2 CPU (full implementation) cache software.	Greenberg	05/24/74	06/17/74	3	

GROUP Multics Development Group DATE 06/26/74 PAGE 5 / 6

PROJECT Performance Improvement Project AREA Performance Improvement Tasks

TASK DESCRIPTION	PERSONNEL	START	FINISH	M-W	CHANGES-STATUS
Backup Speedups					
Design and code new storage system primitive to retrieve initial acls of all rings (for backup).	Mullen	04/13/74	04/27/74	2	Done
Checkout new initial acl primitive.	Mullen	06/10/74	06/17/74	1	
Design and code new storage system primitive to return more status (for backup).	Mullen	06/18/74	06/24/74	1	
Checkout new storage system status primitive.	Mullen	06/25/74	07/02/74	1	
Integrate new storage system primitives into normal backup system.	Mullen	07/03/74	07/09/74	1	

PROJECT Performance Improvement Project AREA Performance Improvement Tasks

TASK DESCRIPTION	PERSONNEL	START	FINISH	M-W	CHANGES-STATUS
System Metering Tasks					
Upgrade metering tools.	Webber	03/01/74	06/01/74	3	DONE
Perform detailed and complete tuning measurements of the system under (controlled?) load.	Webber Roach Jordan	06/15/74	08/01/74	6	