TO: C. Clingen
F. Corbató
R. Feiertag
J. Gintell
N. Morris
R. Roach
J. Saltzer
T. Van Vleck
V. Voydock
S. Webber

FROM: Mike Spier

DATE: October 21, 1971

SUBJECT: Some observations made during the discussions of the new tape management design.

As a byproduct of the recent design effort for Multics' new tape management facility, a number of observations relating to the current Multics implementation was made. This memo is intended to serve as a repository for that information. Following is a list of (possibly unrelated) issues which may deserve some future consideration.

1) It was observed that the implementation of any new facility which manipulates discrete objects (such as tape reels) may eventually snowball into extreme complexity; namely, Multics users which have been accustomed to the segment manipulation flexibility allowed by the file system are expected to demand no less in regards to other proprietary objects. Any such facility may eventually have to give in to user demands and provide niceties such as symbolic names local to some directory, access control lists, renaming etc. It is largely for this reason that the decision was made to use the existing file system services for the new tape reel manager, associating a descriptive segment with each user allocated reel and providing an interface which accepts the segment's pathname as a reel identifier. The only apparent drawback to this scheme is the wasteful usage of an entire page of storage for an a priori relatively insignificant amount of descriptive data. This problem will however be slightly alleviated with the new hardware's smaller page size; moreover, it seems that the unused portion of the page may eventually be utilized for storage of additional information (such as a tape directory) which would have had to be stored someplace, anyway.

?
I don't believe it!

2) An observation relating to the previous one is that even if it were acceptable, from the user interface point of view, to implement a totaly new object management facility (which may be less elegant and refined than the file system) one would still have to solve the associated problems of backup and salvaging. Depending upon the nature of that facility, salvaging might prove critical to the system's well being. The implementation of a dedicated salvager presents of course a considerable additional effort, and because of its necessary interface with the standard salvager as well as other basic system modules may introduce unwanted degrees of complexity and incomprehensibility into the system's supervisor. Here, again, it seems much more reasonable and straightforward to make use of the existing mechanisms.

3) Given the above, it might perhaps be useful to draw a philosophycal line between directory control and segment control, and to consider directory control as a general purpose nonenclature which defines discrete objects, some of which may be segments. This viewpoint in no way affects the file system's implementation, it simply serves to point out a convenient design approach for future implementations.

4) Given the extended usage of segments for control purposes such as the above mentioned tape descriptor segment, care must be taken to prevent off-line movement of such segments. Otherwise, some very interesting deadlock situations may arise, for example an attempt to read in information from some archive tape may fail because that tape reel's descriptor (which is essential for any tape I/O) has been moved off-line and now resides on some other tape reel. Or perhaps a command to move off-line an entire sub-tree may cause the migration of the descriptor of the very tape which is now being written. This in turn indicates that special purpose segments must be distinctly recognizable, either through some naming convention or through a special purpose indicator in their branch.

5) The implementation of any object manipulating subsystem which allows object sharing and which is handled by some dedicated manager other than the file system (e.g., tape reel manager, file manager) requires locking primitives and special salvagers to force-reset a lock belonging to some defunct process. Multics provides only ad hoc solutions to this problem.

6) Objects may be left in inconsistent state following the untimely demise of the manipulating process (e.g., possibly accidentally by a user's leaning on the 2741 on/off switch). An example is a tape which at best has to be rewound, at worst has to be written with a proper trailer, and an allocated tape drive which spontaneously disappears from the list of available devices. The input output device allocation problem has been handled, as a special case, by the IOAM, but the problem at large remains. It would be nice to have a special process termination condition activated in ring-1 of a process prior to its destruction; assuming that ring-1 is immune to user initiated damage, a process termination condition handler might provide a more generalized cleanup facility.

7) The only currently available communication mechanism with the operator is extremely crude and allows only a one way transmission of information, from the user process to the operator's console. There is definite need for a two-way communication capability, for example in the case where the operator does not find the tape reel which he was ordered to mount. Currently, the problem is circumvented by judicious use of the Bell telephone system. This solution may however become increasingly difficult for remote users, especially ARPA network users who may not be in possession of the relevant telephone numbers. The implementation of such a communication facility presents further problems, such as the synchronization of several users who wish to concurrently converse with the operator. Given the ability for a multi-line dialogue it is unclear how the garbling up of the operator's console (intermixing message lines related to different users) can be avoided.