

# THE MITRE CORPORATION

BEDFORD, MASS.

## MEMORANDUM

TO: E. L. Burke DATE: 19 February 1975

FROM: W. L. Schiller MEMO NO.: D73-M-792

SUBJECT: Multics Design Note #11-Revised Top Level Storage Control Specification

COPIES TO: Distribution

This design note documents a revised top level storage control specification. Although a complete specification is attached, most of the functions are relatively unchanged. For unchanged functions, the documentation that appeared in design note #3 will not be repeated. The changes made include the addition of uid's (unique identifiers), a new V-function that identifies a set that corresponds to the model's b, maintenance of records used for directories, a small change in ACL philosophy, and some message segment support.

### uid's

In the initial specification, segments were accessed by temporary, process local names, seg#'s. There was no explicit statement that the information stored in segments was more permanent than the temporary seg#'s, an aspect of the design that should be specified. The use of a permanent, global uid for each segment eliminates the problem.

As before, a parameter to all user functions that read or write segments is seg (or dir\_seg, the # has been dropped). But now the function translates the seg to uid, and all V-functions that refer to segment contents (Branch\_xxx for directories, Uid\_read for data segments) have a uid as a parameter. A segment's uid is kept in the PST, and placed there by Initiate (Figure 2). To make life easier for itself, Initiate only completes the initiation of a segment if it really exists - its alleged parent exists and is a directory, and the segment exists. The Initiate exceptions that a user sees only tell him if his initiations with respect to each other are consistent, not that a segment does or does not exist.

### INAS

The INAS function (in address space, Figure 3) is true if the user is permitted to access a particular segment in a particular mode. The user must have initiated the segment, it must still exist, he must have the appropriate access permission, the security and \*-property conditions must be satisfied, he must be within the appropriate ring brackets (message segments do not have interpreted ring brackets the way directories do), and the type of the segment must be consistent with the desired access mode. INAS is an interpretation of the model's b.

Unlike the model's get access, the specification does not contain any functions that explicitly change the value of INAS. The effects of Initiate, Add\_ACL\_element, Remove\_ACL\_element, Delete, and Set\_ring\_brackets can change the value of INAS for a particular (user, seg, access-mode).

INAS allows the exception conditions for user functions that access segments to be specified more concisely. Also, INAS and the use of uids allows us to specify a current Multics feature that was not in the initial specification and is most easily specified using uid's. Currently, if a segment is deleted and a new segment with the same name and access control attributes is created, users who have initiated the old segment (and had permission to access it) cannot access the new segment without initiating it. This feature was not in the initial specification, but is now specified by having INAS check Uid\_valid, a function set to true by Create, and false by Delete.

#### Records Used

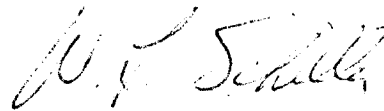
For completeness, exceptions that check for segment overflow and quota overflow have been added to the functions that increase the size of directories, and effect statements that maintain the records used attribute of directories have been added to all functions that change the size of directories. These changes require the use of what I call implementation V-functions (see design note #4). The specific functions used (Figure 24) are NPFA, RPFA, NPFB, and RPFB (new/release page for acle/branch); and ACLE\_offset and Branch\_offset, functions that indicate the actual location in the segment where the new information will be stored.

#### ACL's

Two small changes have been made to ACL support. First, the ADD\_ACL\_element function (Figure #8) uses the acle parameter as the position in the ACL for the new element, rather than using the positioning rules described in design note #3. This change eliminates the functions Find\_position and ACLE\_type. Second, the Add\_ACL\_element function does not check that the access modes in the new ACL element are consistent with the segment type. Instead, INAS checks that the segment type is consistent with the attempted mode of access.

Message Segments

The specification now recognizes "msg\_segment" as a segment type. The Create\_upgraded\_directory function has been changed to Create\_upgraded\_segment. The specification allows any type of segment to be upgraded, therefore any type of segment must be able to have its own quota attribute.



W. L. Schiller  
Intelligence and  
Information Systems

WLS:ms  
Attachment

Distribution: S. R. Ames, Jr.  
D. E. Bell  
E. H. Bensley  
K. J. Biba  
E. L. Burke  
C. S. Chandrasekaran  
M. Gasser  
C. D. Jordan  
L. J. LaPadula  
S. B. Lipner  
J. K. Millen  
R. D. Rhode  
W. L. Schiller  
D. F. Stork  
J. C. C. White