To: Distribution

From: D. R. Vinograd

Date: 01/17/77

Subject: User Notification of Reloaded Segments

Introduction

The present hierarchy reloader and the volume reloader now under development have a common failure; they do not inform a user that the segment he/she is using may not be the one he/she used yesterday, but some prior copy recovered because of a storage failure.

This failure can cause erroneous program behavior when the reloaded data has some relationship to data that has not been reloaded. For example, the recovery of one component of a multi-segment file, but not others, can perturb the MSF in some undefined way. In other uses the recovered file may have a realtionship to some external inputs, such as a memo segment, or a sales list, and if a prior copy is substituted, it may cause grave confusion. To forestall this problem the following mechanism is proposed.

Proposal

As the hierarchy or volume reloader operates, it will turn on a flag in part 3 of the vtoce of any segment it reloads. Since both reloaders create vtoc entries as they operate, no extra overhead is expected. A user settable flag in the entry structure will also be defined. This flag will be settable by an external command, similar to the safety switch commands as well as via the create_branch entry to the append primitive. In the default case, both flags will be off. When the user initiates a segment, a check will be made of the entry flag and, if set, the vtoce flag will be checked. If the vtoce flag is on, initiation will be refused and a non-zero error code and null pointer returned to the caller.

Initiation will be refused until such time as either the branch or vtoce flags have been set to off or the special primitive described below is invoked. Access to modify either the branch or vtoce flags will be the same and will be write permission on the segment or modify permission on the parent.

To allow 'owners' of a segment to check its consistency, a special initiate command will be provided which will initiate the segment even if the vtoce and entry flags are on provided that the caller has the access described above. Thus, the 'owner' of the segment will be able to check the data while 'users' of the segment are still restricted from using it.

To preclude notification of past reloads, turning the flag on in the entry will turn off the flag in the vtoce.

Performance

There should be little or no performance degradation due to reading of part 3 of a vtoce during initiation. This is true for the following reasons:

1. The number of segments with the entry flag should be very few.

2. Of the entries which do have the flag on, many of these will be large components of MSFs and thus will use part 3 of the vtoce during the subsequent activation after initiation.

Extensions

It has been pointed out in discussions that some users of a reloaded segment may not have sufficient access to reset the vtoce flag and, thus, will not be able to use the data until a person with sufficient access resets the vtoce flag. If this proves to be a problem, two solutions are proposed:

1. The special initiate command described above can be modified to only require read access on the segment.

2. A "special" interface to initiate can be created which does not check the entry flag.