

INTERDEPARTMENTAL

MASSACHUSETTS INSTITUTE OF TECHNOLOGY CAMBRIDGE, MASS. 02139

from the office of

February 22, 1971

To: F.J. Corbató
C.T. Clingen
J.W. Gintell
S.H. Webber
M.D. Schroeder

From: J.H. Saltzer

Subject: A proposal for an "unaligned" attribute for segments
in the 645 followon.

Summary

This note proposes a small change in the format and interpretation of the segment descriptor word of the 645 followon processor: an offset field would be included in the SDW, said offset to be added to the word number of any address directed to that segment. This change permits economical implementation of an aligned/unaligned distinction for segments stored by the file system. With appropriate changes to the file system to support this new attribute, one can provide most of the desirable features of the present "archive" command by creating a subdirectory in which all segments are unaligned. The advantage of this approach is that the unaligned segments are all separately "known" to the file system and thus can be accessed just like any other segment, without special case handling.

Motivation

The archive command is intended to save breakage on secondary storage space by packing several segments together into a single segment. Since a segment is allocated secondary storage space in 1024-word records, each separately stored segment has an average of 512 words of breakage in its last block. In general, if one packs N segments together, he will save N/2 storage records. However, once a segment is copied into an archive and the unpacked version is deleted, the file system loses track of the segment. Thus, if it is later brought back out of the archive, the segment will be given new unique identification, and new creation/modification and use dates. An access control list, including ring brackets and modes must be invented

anew. Synonymous names must be remembered and added. The correlation between this segment and older copies on backup tapes is lost because the unique identifier and creation date have changed. Finally, since when in the archive the segment is is not known to the file system, a whole series of mechanisms have been developed to allow the viewing of the archived segment as though it were distinct, and the archive as though it were a subdirectory. These mechanisms are largely duplications of mechanisms which already are (or should be) provided by the file system:

1. The archive command permits
 table of contents, duplicating the list command
 extraction , duplicating the copy command
 update/insertion , duplicating the copy command
2. The print command can print an entire archive
3. The bind command can take as input the contents of an archive

Yet there are restrictions:

1. The PL/I command cannot translate an individual segment from an archive
2. The print command cannot print a single segment in an archive

This proposal suggests that, with the addition of an appropriate feature to the file system to permit automatic packing of segments, the need for a distinct archive mechanism vanishes. The user interface is quite simple: for any segment he creates he may specify the attribute "aligned" or "unaligned", the default being "aligned". If he specifies "unaligned" that segment will be packed together with all other "unaligned" segments contained in the same directory. He may specify that all the segments of a directory be "unaligned" by specifying that attribute for the directory itself.

The user of an "unaligned" segment understands that in return for lower long-term secondary storage costs, he may pay higher cpu charges when he manipulates such segments.

Implementation

The implementation of the "unaligned" segment attribute would be made fairly easy by a small change in the hardware specification for the 645 followon processor: Bits 21-26 of the 2nd word of an SDW are redefined to be an offset which is to be added to the word number before a page table reference is made. The offset field is just large enough

to specify any $0 \text{ mod } 16$ address within one page. (This change shortens the call limiter field so as to permit calls to only the first 512 location instead of the first 16K location of a segment.)

With this change, an unaligned segment can be transferred from secondary storage to core memory in standard page-sized units, without unpacking. If the supervisor sets the offset field and the bounds field in the SDW appropriately, a program can reference such a segment exactly as any other, unaware that the segment is relocated within its pages.

It is presumed that the backup system will treat "unaligned" segments no differently than "aligned" ones except to remember the attribute for later reload. All segments appear as distinct (aligned) entities on the backup tapes.

One would create an "unaligned" segment by declaration; at the time of declaration the file system would automatically copy the segment into the next available unaligned position, starting at a $0 \text{ mod } 16$ address. Deletion of an unaligned segment would result in repacking of the remaining unaligned segments in that directory. Unaligned segments would be packed in groups of 256K words, so that they can use the standard file map/page structure; no segment would be split between two groups.

The "append" attribute requires some special case, since packed segments are not easily appendable. The simplest strategy here is to make the "unaligned" attribute effective only for segments which do not have the append attribute for any user. At the time the "append" attribute is specified, an "unaligned" segment would be unpacked. If the "append" attribute is turned off, the "unaligned" segment could be repacked. A more sophisticated strategy would be to perform the unpacking whenever an "unaligned", "appendable" segment is mapped into some address space, and repacking when it is terminated.

Coordination of shared packed segments

This proposal appears to have some implications for the system segment table organization, but as far as I have been able to see, the impact is minimal. If two address spaces both contain the same unaligned segment, they each contain an SDW with the same absolute page table address, offset, and bound field. In other words, the page table/file map for the entire packed group is shared as usual. Access control is in the SDW and thus not shared. If two address spaces contain two unaligned segments from the same group, the single file map/page table for the entire group of packed segments is still used; distinct SDW's point to different parts of the page table, and contain different offsets and bounds fields. Packing and repacking of unaligned segments would be accomplished by taking all active SDW's affected temporarily out of service, repacking the segment, and then restoring the SDW's with appropriate changes.