

TO: Distribution
FROM: Steve Webber
DATE: April 28, 1972
SUBJECT: Simulating BOS PATCH for PML Systems

The following memo describes some of the structure of page multilevel's data bases as well as gives instructions on how to **Patch (in BOS) to keep copies up-to-date, etc.**

General Required Action

Active Segment

Page on drum

In core - turn on pdme.mod (ok if going)

Not in core - turn on pdme.mod

Not on drum

In core - ok

Not in core - ok

Not Active

On drum - turn on pdme.mod, patch drum

Not on drum - ok

Given disk address, to find if a given page is on the drum do the following:

1. Dump drum record 20(8). This is the pd hash table. (Record 10(8) if two-fold dim.)
2. Take low order 10(10) (9(10) if two fold dim) bits of disk address. Split this into low order bit and other bits. Use 'other' bits as **index into hash table** -- this finds a word. If the low order bit (above) is a 0 use the left half of this word. Otherwise use the right half. The value in the selected half is a "hash thread pointer" which points to a threaded list of entries in the pd map. (The thread is in the right half of word 2 (starting at 0) of each pd map entry.) Follow this thread searching for a map entry with the device address for the given disk record. The device address is in the left 22 bits of word 1 of a pd map entry. If an entry is found, the relative offset of the entry can be used to find the drum address -- divide the address of the map entry by 4 -- this gives the drum address. If an entry is not found (the device address was not found in the thread) it does not have a copy on the drum.

When following the thread, the thread pointers are word offsets into the pd map. The pd map occupies the first 20(8) (or 10(8), etc.) records of the drum.

Given a segment number and offset, the following can be done to find all copies of a given page if the segment is active (if the segment is not active use disk address strategy above after finding the appropriate file map -- only

disk addresses will appear in directory file maps.)

1. Find the aste/pt for the segment (either by using its SDW or by using astep in branch, etc.).
2. Find the ptw for the page of interest.
3. If the access field of the ptw (right 6 bits) is a 64(8) the page is in core. To find the core map entry take the left half of the ptw (core address mod 64) and divide this by 4. This will then be the offset of the cme in the core map. Add to this offset the offset of the base of the core map, 1000(8). This is a relative pointer to the cme within the sst. The left 22 bits of word 1 (starting at 0) of the cme is the device address.
4. If the access field of the ptw is 10(8) the page is not in core and the left most 22 bits of the ptw is the device address (devadd).
5. The right 4 bits of the device address is the device id (did). If this is "1001"b, the devadd is for the pd and the most recent copy of the page is on the drum. If this is the case, use the left 18 bits of devadd, the drum record number, to find the pd map entry. Multiply the drum record number by 4, add the offset of the pdme in the sst. sst (4000) to come up with a relative pointer to the pd me in the sst. (If you do not add the 4000, the pdme can be found directly on the drum. However, if there is an sst use it.) The left 22 bits of word 1 (starting at 0) in the pdme is the disk device address.

6. The final device address found can be used for patching. However, if there is a copy of the page on the drum, only the drum copy need be patched if the following patch to the appropriate pdme is made.
7. If a drum page is patched the corresponding pdme should also be patched -- both in the sst and on the drum. The "modified" bit of the pdme (000000004000) is the word 1 of the pdme. It should be turned on. (It may already be on.)