To: Distribution
From: M.A.Padlipsky
Subject: Meetings 1 & 2
Date: 3/24/72

Aside from the attempt to solve critical bugs, Gintell suggests that the group should also concern itself with techniques to lower the percentage of "unexplained" crashes.

PML was the major concern. 1) Symptoms: The Paging Device Map becomes threaded into multiple, self-consistent sublists instead of one big list; no free entries can be found, eventually; PML "shuts off". 2) Observations: Webber notes that 10-12 at night is frequent problem time; this might be because of the traffic Catchup causes. Daley suspects threading may be recursively invoked; could be lock failure or an interrupt coming through despite masking. The problem has not yet happened on a 1 cpu system. Snyder and Webber have gone over the threading code closely and a programming bug is unlikely; Morris and Daley will also eyeball the code if need be. 3) Actions: Add checking code to verify that a free PDM entry exists each time through, check whole list every 100 times. Decrease size of paging device to increase traffic and force problem so as to get traps to go off, and to see if it will happen on 1 cpu. Run 1 cpu hard to attempt to rule out 2 cpu problem as cause (switch cpu's to see if it's cpu-specific). Place an "in use" switch in the threading routine to see if it's being re-entered. Record last time manipulated in PDM entry. Webber will put together a special system containg all the traps, for use a.s.a.p.; some 1 cpu stressing will be done with it, but getting the traps to go off (hence 2 cpu operation) is more important.

Null PTWs were discussed briefly. Webber proposes to have each routine which generates a null address place a code so that we can determine who's responsible. Consensus was that this is a useful course, problem put aside until it's been done.

Locking errors were discussed at some length. 1) Symptoms: lock does not contain process id of locking process when attempting to unlock; lock contains pid after unlocking. 2) Observations: Mishandling of out of service bit (because of reads being queued before writes) or modified bit could be the culprit. Page trace has shown faults on pages 0,1,0 of the IOAT (lock in work 0) in 147 ms. Problem frequently happens right after loading. Webber has a strong suspicion pre-page / post-purge is involved. Possible STAC failure should not be ignored. 3) Actions: Wiring the IOAT should minimize the problem, as most crashes occur on references to its lock; this gives us breathing space to fix PML, work out more traps here; this will be done in special PML system. Jordan's new lock may cast some light; it will be installed after special session 2 cpu test. Thought that rumored misbehavior of index register 0 might be involved tended to die out after airing of rumor that PL/I still hits op not completes after changing to X1. Pre-page / post-purge will be turned off Moday to see if Webber's hunch is right, unless other experiments seem more fruitful.

The "spontaneous" setting of date/time modified was attributed to the hardware's turning on the modified bit before access checking. This was confirmed after the Friday meeting. Ohlin has been asked to investigate.

Bad ASTE trailers and bad SDW's were also discussed, but I didn't hear/understand any recommended actions other than Roach's suggestion that ignoring the bad trailers might be a bad idea; consensus was it's better to have to retrieve some files for *a* user than crash the system on everybody (the alternative to ignoring). These problems are thought to be related to locking, in that "backing up a page in time" (bringing in a fresh copy of a page which had been modiifed) might account for everything. This could be very significant, but I don't really follow it yet.

¶ There will be a meeting at 2 Saturday to assess PML special system behavior, decide how to run it during the rest of the weekend.