

PROGRAMMING STAFF NOTE 55

FROM: M. Padlipsky
SUBJ: What We Learned - Or Should Have Learned -
From the Hardware Transient Hunt
DATE: November 23, 1965

Introduction

As a result of my experiences in the role of co-investigator of the suspected hardware transient situation at the Computation Center, I feel that certain lessons become clear (some more painfully than others) which we all would do well to think about. At the risk of sounding preachy, the "Great C.H.T. Hunt" (C.H.T. = Corby's Hardware Transient) would have been far less trying had we squarely faced some known problems long before it started; and, when the C.H.T.'s (Ø = Other) appear, whoever goes off to hunt them will benefit from our taking certain actions now. We did catch the C.H.T., finally, but

Communications

Through one of those cosmic coincidences which make the world go, three reports about possible trouble with SET/NET on the Center's 7094 were finally received by the hardware transient investigators within an hour of our discovery that the failure of those two instructions was indeed "the" hardware transient. The sources were:

- 1) a phone call to one of the Center staff from a programmer in the Institute's Records Processing Section to the effect that RPS "hadn't used A-core for months" because of apparent problems with zero tests;
- 2) a chance encounter with a MAC programmer who, when informed of the discovery, replied that he had had similar problems in the past but "had never been able to prove it";
- and 3) most embarrassing of all, the hazy recollection by another member of the Center staff of an unofficial conversation which implied that a non-departed colleague might have been assigned the specific

task of investigating the ZET in the Spring of this year. ("Unofficial conversation" = "I think I overheard, once".) Another interesting background point is that the final resolution - and authorization - to perform our test of the suspect instructions was reached in large measure because attendees of a MAC staff meeting recalled having encountered the same problem some years ago.

Perhaps the above is all that need be said. However, it should be beneficial to attempt to make explicit the "obvious" lessons. The main point is that we could have been spared a good deal of work (to say nothing of machine time) had any of the four pieces of information been made available at the beginning of the "hunt". Therefore: a) It would seem that (= "please,") programming staff meetings at the Center should as a matter of course emphasize feedback on problems, especially because of the large amount of contact the consultants have with users. b) It would also seem that feedback should be actively encouraged, not only from users but also between the Center and MAC staffs; perhaps channels for this could be developed - or if such channels exist already, be publicized. c) We have communication problems. We now also have a very clear precedent for trying to cope with them on an organized basis. d) At least we could have some arguments about whether anything can be done - which would, of course, itself be doing something to cure matters.

Operations

In the general area of Operations, one point became quite clear and one other emerged vaguely. It is clear that the present philosophy implicit in Operating Staff Notes is not tenable unless the authors of the programs the Notes are about are available to the operators. Center Operators, not having had the programmers in their machine room day after day for test sessions as their MAC counterparts did, paid the price levied by too-general, ambiguous, rather programmer-oriented documentation. A remedy is being worked on in terms of a series of

highly-detailed instruction sheets for operation of current programs; this approach should work and it would seem that all future Operating Staff Notes should embody the step-by-step rather than the "big picture" philosophy. Indeed, "Instruction Sheet" seems to have more useful connotations in this context than "Operating Staff Note". The favorable effects on Operating Staff morale should be sufficient justification for this view, and better performance is also a likely result. (Multics documentation might also benefit, especially if the system is intended to become of more-than-local utility.)

Somewhat less clear is the lesson in the area of programmer operation. Certainly, a system programmer who is also a virtuoso at the operating consoles is one of the best kinds of magician to have around; and despite the consideration that virtuosity cannot be learned, a case might very well be made for offering interested systems programmers training in operating the machine. Efficiency and morale are also factors here, and again Multics (and even the 360 workers) might benefit - it's a very good thing to have resident magicians around.

Maintenance

Another useful coincidence (no wonder all the lights went out later - all the day's good luck had been spent) turned up when the cards on which the SET/NET test program had been punched were being listed. The 407's devouring one card and decapitating a second served to underline the fact that we had been, after all and for all that long, fighting hardware problems. Perhaps the successful resolution of the "hunt" offers a lever through which some moral suasion could be applied to IBM, to encourage more painstaking maintenance work at the Center. (The tape drive problems seem chronic.) At any rate, the Staff are more than welcome to use the C.H.T. tale on any recalcitrant hardware types they know.

Configuration

A final lesson emerged from the deluge of print out and hastily - scribbled-cards-with-machine-conditions-on-them: Poloroid cameras should be mounted in front of the consoles to record the lights when the machines stop. Though this is the simplest lesson, it would seem to be the most practicable of the solutions.