

Thoughts About Operating Multics

Don Oppert

December 17, 1965

Most of what follows is (or should be) obvious to anyone who has had contact with computers. Yet, I doubt if any good (i.e., effective and efficient) operating system today contains all.

1. No operating-instructions memo (OIM) should ever go to the operations crew without follow-up. That is, operations people are not going to admit (until pinned to the wall after a catastrophe) that their instructions were incomprehensible. Every OIM should be dated, signed, and the system module with which the OIM is associated should be identified; e.g., XYZ2L3.

2. No verbal operating instructions should ever stand alone (i.e., without written support) for longer than one test session. It's the operator's word against the programmer's and an operations-oriented computing center manager will always back the operator in this kind of dispute.

3. No OIM should ever exceed 5 pages! Sounds crazy but so does transparent code. Reasons are obvious.

4. Operators are not machines! Obvious, yet many OIM's read like complex flow-charts. They must not, if recovery from catastrophe is expected at 3 A.M. as well as any other time.

5. On line instructions (from console typr or printer) as well as OIM's must never be vague or general in terms of time. That is, an operator should not be instructed to depress a key "for a short time" or "momentarily", etc. He should be told when to set and then told when to reset.

5.5 And every on-line message should be "signed". That is, the section of the system sending the message should be identified.

6. Whenever an operator is required to enter data into the system, such as resetting a clock, or supplying a system name, there should be an echo check of his input.

7. Operators should be included in Test Sessions. Not only is their training less painful, they can and will point out defects in operational procedures. Correcting these becomes part of the test sessions and potential difficulties are avoided.

8. When the "impossible" happens in a program, i.e., for which the programmer has inserted: HTR * , there will be trouble. The operator will assume such a case is not covered in any OIM and won't find pertinent instructions unless they are literally under his nose. Consequently, the machine will probably idle for a while and then a dump (hopefully) will be taken followed by a reload of the system. Moral obvious: never, never, never put an HTR * in a program which you yourself will not always operate. At least put in a target address which will, if at all possible, resolve the difficulty and resume operation of the system. Why couldn't there be a target address in the supervisor which produces a dump? The call could give first and last word addresses to be dumped as well as the type of dump.

9. It is not sufficient merely to tell operators how switches are to be set. When this practice is followed the maintenance man will frequently be called only to learn a switch has been set incorrectly. However, if the operator knows the function of a switch, not only is he more apt to check its setting but might be able to diagnose trouble resulting from a bad setting.

10. Operators should be given token allocations of resources. It is more fun to operate a system when one knows something about the something he is operating. Suggestions to improve operational procedures, higher morale, and more efficient operation will be the result.

11. There should be a display console near the computer. Not only would the proximity of such a device aid in debugging system programs, operating personnel could perform controlled experiments and/or make observations of system behavior at all hours and under all conditions.