T+D in Multics   Outline of ideas.   J. H. Saltzer, Mar. 10, 1966

General Pattern: Initial T+D will be primitive

a. Use available T+D programs

b. Take system down when necessary

Later ~~will~~ can be sophisticated

a. Special integrated T+D programs

b. System rarely, if ever, down for T+D.

What does Multics have to offer the T+D programmer already?

1. Reconfiguration + Partition. While the system is running, it can "pause" and reconfigure ~~to stop using~~ avoiding Drum module, certain hardware (Memory Controller, Processor, etc.) By throwing switches, the disconnected units can be turned into a standalone computer and run Bedfort T+D programs. This T+D is done simultaneously with Multics operation, but independently. (Multics ~~is running~~ runs "slow" in the interim)

2. Resource assignment. While the system is running, ~~it can assign to users~~ any user can request assignment of the exclusive use of an I/O device (typewriter, tape drive, card punch) on a sign-up basis. The T+D programmer can write a program which operates as any other process in Multics, and exercises the ~~assigned~~ device it has signed out. This concept can even be

extended to processes without reconfiguration for those tasks which can be run in slave mode and can withstand interrupts.

3. ~~Demon~~ Periodic checkout without human intervention. The T+D programmer can create a "demon" process which, using the real-time interval clock, will wake up periodically, perform some checks, and go back to sleep if everything is OK. ~~This~~ Technique could be applied, for example, by having the ~~process~~ process ~~reset~~ An interesting extension of this technique is to have the demon, on its last action of each period, reset an alarm clock which will warn the operator if the demon ~~has never~~ does not wake up ~~again~~ in a reasonable time.

~~If desired, such a demon process~~

4. Special Authority. Multics has provision for administratively assigned authority to perform certain operations not normally allowed. For example, the authority to pre-empt ~~the~~ a ~~processing system for~~ type class which is setting suspicion might be granted to a process belonging to T. + D. ~~operator~~ engineer. Although such authorities are easily extendable, there is the difficulty that ever a T. + D. routine cannot be allowed, though a programming bug or deliberate bothouse error, to "bring down the system". Safeguards to system security must also be applied to T. + D. ~~user~~ processes.

5. Multilevel Storage Management. For cases where, for example, exhaustive checks of a Drum module must be made, the Multilevel Storage Management routines can be requested to remove user files from the area in question. Scroll A request must be made with appropriate authority, and well in advance, since considerable effort may need to be expended to accomplish unloading of a module. (Note the necessity that the stored files be accessible to their owners at all times, and that the contents of the files should not (?) be divulged, even to the T.+D. person.)

6. System error logs.