PUBLISHED: 9/13/66

Identification.

Overview of Fault Handling
Chester Jones

Purpose.

Section BK.3.01 contains an overview of fault handling in Multics. Sections BK.3.02 through BK.3.09 contain the detailed specifications of the individual modules and data bases.

References.

A summary of the major goals of Multics fault handling is contained in Section 11 of "Structure of the Multics Monitor for the GE 645," by V. A. Vyssotsky, F. J. Corbató, and R. M. Graham, Fall Joint Computer Conference, 1965. Although much of that paper has been superceded by sections of this manual, it is still useful for background information and motivation. It is important to note that Multics executes as part of the user process and that each user process is divided into a number of mutually exclusive subsets, called rings. User process is defined in Section BJ.1, Process Control; rings are described in Section BD.9, Protection of the Supervisor. It is recommended that both Section BJ.1 and Section BD.9 be read before Section BK.3. The design of Multics fault handling is motivated, in part, by the design of the GE-645 processor. At present, General Electric Computer Department publication number M50EB00107, "Engineering Product Specification, GE-645 Processor," serves as the processor reference manual.

Introduction.

A primary design objective of Multics fault handling is to allow a user process to supply a procedure for handling any fault that is not reserved for the operating system provided the user process has administrative authorization for handling that fault. The Multics operating system contains a standard procedure for handling each fault. However, for each fault that is not reserved for the operating system, there is at least one point in the fault handling procedure where control may pass to some other procedure in the process if that process is administratively entitled to provide alternative handling for that fault.

We define four general classifications of faults; program generated, operating system generated, hardware generated, and manually generated. Each fault is placed in the classification that corresponds to the possible cause of the fault. If a fault could only be caused by a hardware malfunction, it is classified as hardware generated; if it could possibly be caused by the execution of a program, it is classified as program (or operating system) generated. All hardware and manually generated and certain operating system generated faults are called system

faults. They may occur at any time, regardless of which user process has control of the processor. They may not, in general, be attributed to the running process. The memory parity fault, the startup fault, and the illegal descriptor fault are examples of system faults. Certain of the program generated and certain of the operating system generated faults have reserved meanings to the Multics operating system regardless of when they occur. Such faults are called reserved faults. For example, directed fault 0 is reserved to indicate a missing page or segment; the connect fault is reserved to mean "clear your associative memory." The remaining program generated and operating system generated faults are called user faults. Since Multics executes as part of the user process, all user faults may be attributed to the running process. Overflow, derail, and fault tag 2 (linkage fault) are examples of user faults. Complete lists of user faults, system faults, and reserved faults are contained in Section BK.3.02.

Multics fault handling consists of two parts which perform the following functions:

1. Replaces the handler for any fault condition that is not reserved for the operating system upon request from a user process that has administrative authorization for handling that fault.

2. Serves as the interface between the hardware and the procedures for handling a fault condition when that fault condition occurs.

The extent to which a user prccess may provide replacement fault handling procedures is controlled by the system administrator; it depends on the possible effect the replacement procedures may have on the overall operation of the system. Procedures for handling reserved faults are built into Multics and may not be replaced while the system is running. Procedures for handling system faults affect the operation of the entire system since they are shared by all user processes running under the same version of Multics. The effect on the overall operation of the system, that procedures for handling user faults may have, varies, since some parts of the operating system are more sensitive than others. (See Section BD.9, Protection of the Supervisor.) For example, procedures for handling overflow faults that occur within the user program's domain of access can affect the operation of only a single user process; procedures for handling overflow faults that occur in the hard core ring can affect the operation of the entire system since all of the shared data bases are "exposed" while control is in the hard core ring.

Figure 1 is a rough block diagram of the modules and data bases for handling faults in Multics. The solid lines indicate the flow of control between modules; the dashed lines indicate the flow of data between modules and data bases.

All of the modules involved in Multics fault handling execute as part of the user process whenever that process requires some action regarding fault handling. These modules are the fault maintenance module, the catastrophe module, the fault interceptor, and the individual modules for handling the various faults. Only the fault maintenance module may be called by procedures not shown in the diagram; the remaining modules are invoked as a result of a hardware fault condition.

Fault Maintenance Module.

For each protection ring of each user process, the fault maintenance module (Section BK.3.05) maintains a list of pointers to the procedures for handling user faults that occur in that ring. In addition, the fault maintenance module maintains a system-wide list of pointers to the procedures for handling system faults. When a user process is created (Section BJ.10), it is christened with lists of pointers to the "standard action" procedures for handling user faults in the respective rings. All user processes running under the same version of Multics share the procedures for handling system faults. In order to replace the procedure for handling any fault that is not reserved for the operating system, the user process calls the fault maintenance module, passing the name of the fault condition and a pointer to the replacement fault handling procedure. The fault maintenance module determines whether the caller is administratively entitled to provide a replacement procedure for handling that fault, and, if so, places the pointer to the replacement procedure in the appropriate list.

Fault Interceptor.

The fault interceptor (Section BK.3.03) is a master mode procedure that can be entered only as a result of a hardware fault condition. Since the fault interceptor contains the wall crossing mechanism (Section BD.9), it is accessible in every ring of each user process in the sense that it can be entered without first crossing a wall, and it is able to switch rings when necessary.

When a Multics processor generates a fault, control passes automatically to the fault interceptor which executes as part of the process that is running at the time of the fault. While executing within the ring in which the fault occurs, the fault interceptor safe-stores the processor state in the Process Concealed Stack (Section BJ.1.05) that belongs to the running process and makes space available for safe-storing the processor state should another fault occur. Then, the actions taken by the fault interceptor vary, depending on the probable cause of the fault. For missing-page faults, the fault interceptor switches to the hard core ring descriptor segment and uses the Process Concealed Stack to call the Basic File System (Section BG) to supply the missing page. For other system faults, the fault interceptor switches to the hard core ring descriptor segment,

copies the safe-stored processor state from the Process Concealed Stack into the hard core ring stack that belongs to the running process, and calls the appropriate procedure for handling the fault. For user faults, the fault interceptor copies the safe-stored processor state from the Process Concealed Stack into the stack for the ring in which the fault occurred and uses that stack to call the procedure for handling the fault in that ring. It is important to note that for user faults, the fault interceptor does not cross a wall, but executes entirely within the ring in which the fault occurs. If, and when, control returns from the fault handling procedure, the fault interceptor checks the validity of the processor state, restores the processor state, and returns control to the point at which the fault occurred.

Catastrophe Module.

The catastrophe module (Section BK.3.04) performs the initial handling of system faults that indicate either an appending hardware malfunction or some possibly fatal error in the operating system. For example, following an illegal descriptor fault, the catastrophe module checks the validity of the descriptor for the fault interceptor before transferring control to the fault interceptor.

The catastrophe module is an absolute mode module that can be entered only as a result of a hardware error condition.

Ground Rules for Fault Handling Procedures.

The following is a set of ground rules for both user-supplied and "standard action" fault handling procedures.

1. Each fault handling procedure may modify the safe-stored control unit information if some modification is required. The fault interceptor does not modify the control unit information. The fault interceptor does check the validity of the safe-stored processor state after control returns from the fault handling procedure.

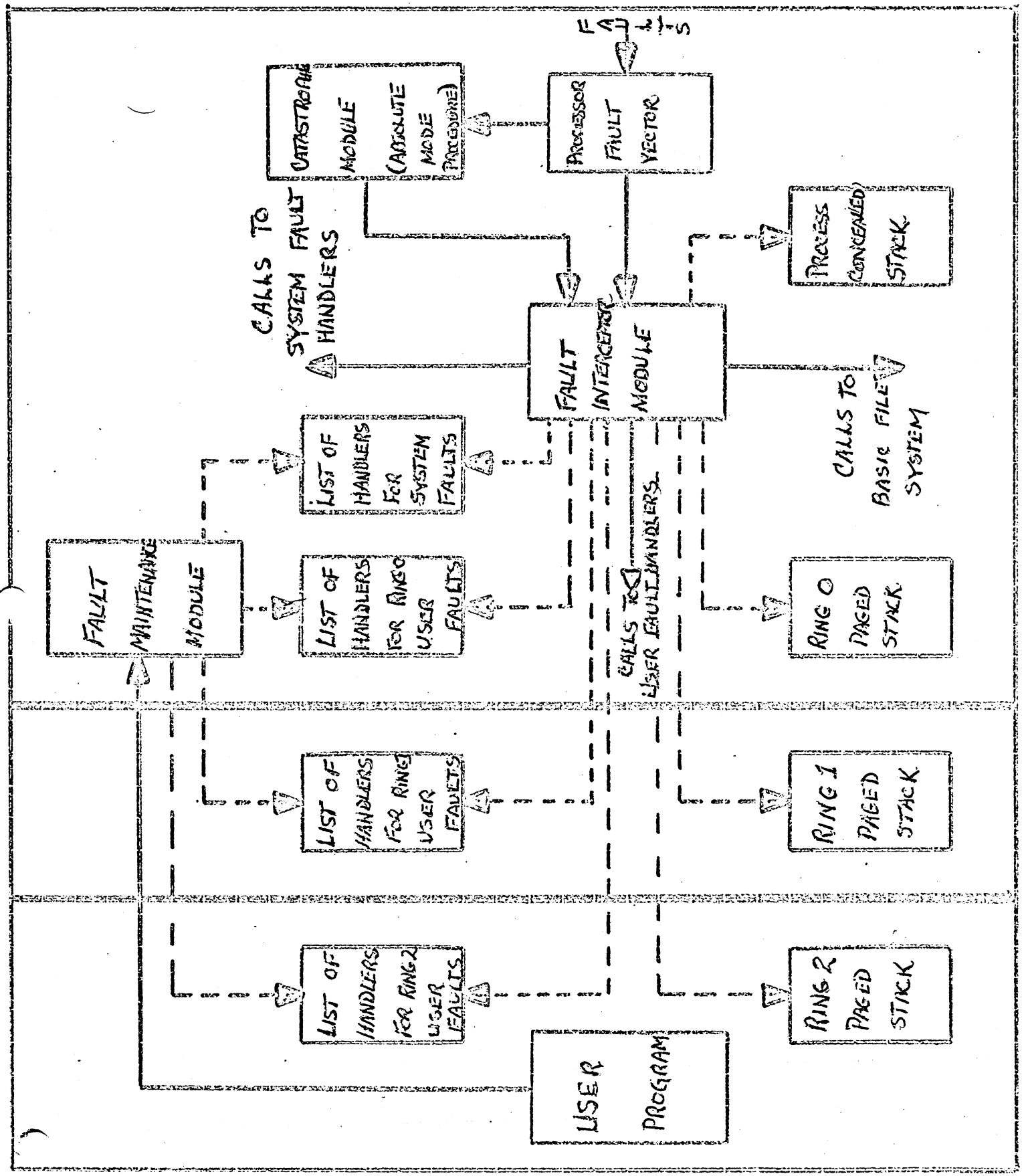
FAULTS
CATASTROPHE MODULE (ABSOLUTE MODE PROGRAM)
PROCESSOR FAULT VECTOR
CALLS TO SYSTEM FAULT HANDLERS
PROCESS (CONCEALED) STACK
FAULT INTERCEPTOR MODULE
CALLS TO BASIC FILE SYSTEM
LIST OF HANDLERS FOR SYSTEM FAULTS
FAULT MAINTENANCE MODULE
LIST OF HANDLERS FOR RING 0 USER FAULTS
CALLS TO USER FAULT HANDLERS
RING 0 PAGED STACK
LIST OF HANDLERS FOR RING 1 USER FAULTS
RING 1 PAGED STACK
LIST OF HANDLERS FOR RING 2 USER FAULTS
RING 2 PAGED STACK
USER PROGRAM

FIGURE 1