Published: 10/13/66

Identification

Overview of Processor Management
Chester Jones

Purpose

Section BK.0 contains an overview of processor management in Multics. Processor management consists of two parts, interrupt handling and fault handling, which are described more completely in Section BK.2 and BK.3, respectively.

Discussion

There are two basic events which cause a processor to depart from the procedure it was executing. A condition detected within the processor hardware (e.g. accumulator overflow) causes a processor fault. The conditions which cause faults are not necessarily error conditions. For example, a program reference to an area not currently in core memory causes a "missing-page" fault. A signal from some source other than a condition detected within the processor hardware causes a processor interrupt. An interrupt may be triggered, for example, by an I/O device or by a processor. A processor may delay recognition of certain faults and of all interrupts by operating in inhibited mode.

The processor response is the same for a fault or an interrupt; the processor stops what it was doing and executes a unique pair of instructions associated with that specific fault or interrupt. The instruction pairs associated with faults constitute the fault vector; the instruction pairs associated with interrupts constitute the interrupt vector. When a Multics system is initialized or reconfigured, the fault and interrupt vector pairs are set to store the processor state in special segments and transfer to the appropriate procedure for servicing the fault or interrupt. At some point, after the fault or interrupt has been serviced, the processor state is restored and control is returned to the point at which the interruption occurred.

In Multics, a distinction is made between interrupts caused by a processor operating in the Traffic Controller modules of the supervisor and interrupts caused in some other manner. Interrupts that are caused by a processor operating

in the Traffic Controller modules are called process interrupts. They are directed to the process currently in execution on the processor being interrupted. System interrupts, on the other hand, are caused by other devices. They are directed to some process in the system, generally not the one on the processor being interrupted. Similarly, a distinction is made between faults that indicate hardware malfunction and faults that are caused by the execution of a program. Program generated faults are called process faults; they are caused by the user process in execution. Faults that indicate hardware malfunction are called system faults. They are directed to some process in the system, but generally not the one executing on the processor detecting the fault.

Gross Division

The various procedures and data bases involved in Multics processor management fall into several basic groups. In the first group are the Interrupt Interceptor Module (IIM) and the Fault Interceptor Module (FIM). All interrupts are passed directly from the hardware to the IIM, all faults are passed directly from the hardware to the FIM. The IIM and FIM are standard (re-entrant) Multics procedure segments. They execute in master mode, but not in absolute mode. However, they are unique with respect to their unorthodox method of entry. Unlike all other Multics procedure segments, the IIM and FIM are not called, but are entered directly from the hardware. Since they are not called, the IIM and FIM must establish their own stack and linkage pointers. Finally, the IIM and FIM differ from other Multics procedure segments with regard to their mode of protection. (See MSPM Section BD.9 for a description of the Multics protection mechanism.) While other Multics procedure segments are usually accessible as procedures in only one protection ring, the IIM and FIM are accessible as procedure segments in every protection ring of every user process.

Functionally, the IIM and FIM are quite simple. They save the processor state in a safe place and call a handler to service the interrupt or fault. When control returns from the handler, the IIM or FIM restores the processor state and returns control to the interrupted program.

The second group of procedures includes the handlers for servicing the various faults and interrupts. The handlers are standard slave-mode procedures; they take whatever action is indicated by the fault or interrupt and return control to the caller. Handlers for system interrupts

are brief and to the point. They cannot contain programmed process faults such as missing-page faults. They decode the meanings of the interrupts and notify the appropriate processes. Handlers for process interrupts exercise the primitives provided by the Traffic Controller. For example, the handler for the time-out interrupt calls the Restart entry of the Traffic Controller. Faults are handled, largely, by other Multics modules. For example, missing-page and missing-segment faults are handled by the Basic File System (MSPM Section BG).

Stack Usage

When a Multics system is initialized or reconfigured, a special stack is created for each processor available to the system. This stack, called the Processor Stack, is retained in core memory as long as the system is running. It is used by the Interrupt Interceptor Module and by the Fault Interceptor Module following a system interrupt or a system fault to store the processor state and to call the appropriate handler for servicing the fault or interrupt. Each of the respective Processor Stacks is referenced by a common segment number; when a processor is switched from one process to another, the Processor Stack for that processor is passed along to the next process.

When a user process is created, it is christened with a special stack called the Process Concealed Stack. The Process Concealed Stack is retained in core memory while its owner process is running. It is used for safe-storing the processor state following a process interrupt or a process fault and for calling the handlers for servicing process interrupt and process faults. The process Concealed Stacks for all processes running under the same version of Multics are referenced by a common segment number.

GE-645 Hardware Considerations

The GE-645 system may include a maximum of eight active devices (processors, drum controllers, and GIOC's) and eight system controllers. Each processor operates internally independent of the other processors. All communication and information exchange occurs between a processor and system controller. Each system controller has a manual switch which identifies the processor to which interrupt signals are sent by that system controller. The processor so identified is called the control processor for that system controller. One processor may be designated control processor for more than one system controller, but one system controller may have only one control processor.

Each system controller contains 32 execute interrupt cells and a mask register for each of the cells. When an execute interrupt cell is set on by same active device and the associated mask register enables its recognition, the system controller sends an interrupt signal to its control processor. Each of the 32 execute interrupt cells has a corresponding pair of instructions, the interrupt vector pair, which the processor executes automatically when it receives the interrupt signal. The 32 interrupt vector pairs form the interrupt vector for that system controller.

The GE-645 processor fault repertoire includes 32 distinct fault conditions. Corresponding to each of the 32 fault conditions is a pair of instructions, the fault vector pair, which the processor executes automatically when it generates the fault. The 32 fault vector pairs form the fault vector, a 64-word block of core memory whose base address is determined by switch settings on the processor maintenance panel.

When the GE-645 processor control unit detects a fault condition or accepts an interrupt signal, it takes a "snapshot" of its internal status and executes the fault or interrupt vector pair that corresponds to the condition causing the fault or interrupt. When a Multics system is initialized, the fault and interrupt vector pairs are set to store the processor status in either the Processor Stack or Process Concealed Stack and to transfer to the Interrupt Interceptor Module or to the Fault Interceptor Module, as appropriate.