Published: 11/17/67

Identification

Hardcore ring unrecoverable condition handler (trouble module)
J. H. Saltzer

Purpose

The trouble module is provided to allow a standard response to unexpected and unrecoverable conditions detected within the hard-core ring. Throughout the Multics supervisor, a pattern of redundancy and cross-checking is an integral part of the design. In early versions of Multics, the precise action to be taken upon failure of a redundancy check may not be apparent, and many failures will be due to programming errors. Such failures detected in the hardcore ring are therefore handled by a direct call to the hardcore ring unrecoverable condition handler. This procedure brings the system to an orderly stop as quickly as possible so that diagnostic debugging is possible.

As experience is gained with Multics operation, and confidence in hardcore ring procedures increases, more appropriate recovery procedures are expected to be inserted for many conditions which are initially handled by the trouble module.

Implementation

(This implementation description is functional and brief, and will be replaced by a more complete design specification.)

The calling sequence to this procedure is:

```
     dcl trouble_code fixed binary(17);

     call trouble(trouble_code);
```

where trouble_code is an integer unique to this particular trouble.

The following steps are taken by the trouble module:

1. Inhibit, then mask all interrupts.

2. Store the processor state in this CPU's storage slot in the trouble module.

3. Mark this processor as "checked in".

4. Check trouble lock to see if another processor has detected the error already and is shutting down the system. If <u>on</u>, skip to step <u>9</u>.

5. Send an emergency interrupt to all processors not shown as "checked in". The emergency interrupt cell is never masked during normal Multics operation. The emergency interrupt handler calls the trouble module. Thus the operator may force a trouble call by setting the emergency interrupt cell.

6. Shut down all GIOC channels, saving current status of each channel.

7. Shut down the Drum controller, saving its status.

8. Reset the calendar clock alarm register.

9. Set up standard alarm indication (e.g., trigger the GIOC alarm bell and set a pattern in CPU lights, which includes the trouble code described below and the segment number of the procedure which called "trouble".)

10. Stop on a DIS instruction.

<u>Note</u>:

The initial implementations of many hardcore ring modules include a call to a trouble handler named "panic", without arguments. As the hardcore ring modules are reviewed and upgraded, calls to "panic" should be replaced by calls to "trouble" and appropriate trouble codes should be defined. The procedure "panic" is implemented as a call to "trouble" with trouble code = 1.

<u>Trouble Codes</u>

Below is a list of all assigned trouble codes (in octal), the name of the segment which will call with that code, and the probable meaning of the particular error condition found.

1. Call to procedure <u>panic</u>. A variety of hardcore ring modules call this entry, with a variety of meanings.