

PUBLISHED: 9/13/66

Identification.

The Connect Fault Handler
Chester Jones

Purpose.

When it receives a connect signal from another processor, the GE-645 processor generates a connect fault. In Multics, the connect fault is reserved to mean "clear your associative memory." The connect fault handler is the procedure in the Fault Interceptor Module which is executed in response to a connect fault. The connect fault handler is shared by all processes running under the same version of Multics and is executed entirely in master mode with all interrupts inhibited.

Discussion.

It is essential that certain Multics modules be able to clear the associative memories of all processors in the system. For example, when the Basic File System (Section BG) removes a page or segment from core memory, it must clear the associative memories of all processors in the system. A program executing on one processor cannot clear the associative memory that belongs to another processor; it can only "ask" the program executing on the target processor to clear that processor's associative memory. It accomplishes this by sending a connect signal to the target processor.

Connect_generator is a master mode procedure which issues connect signals to all processors in the system. Connect_generator is accessible to those Multics modules which must clear the associative memories of all processors in the system. Connect_generator is called with no arguments as follows:

```
call connect_generator
```

For each processor available to the system, connect_generator performs the following steps:

1. Sets a flag to non-zero for the target processor. Using the processor index number as an index into the connect flag array, the connect_generator sets the entry for the target processor to non-zero. (See Section BK.1.04 for a description of the connect flag array.)
2. Issues a connect signal to the target processor. Using the processor index number as an index into the connect operand word array, the connect_generator executes a connect instruction (cioc) whose address points to the base location of the connect operand array in the Processor Communication Table (Section BK.1.04).

3. Waits for a positive response from the target processor. When the target processor detects the connect signal, it will respond by clearing its entry in the connect flag array of the Processor Communication Table.

(In the implementation of connect_generator, Steps 1 and 2 are repeated for every processor available to the system before Step 3 is executed.)

Actions of the Connect Fault Handler.

When a Multics processor receives a connect signal, control passes automatically to the connect fault handler which executes on behalf of the process that is running at the instant of the fault. The actions of the connect fault handler are as follows:

1. Temporarily stores the processor state in the Process Concealed Stack. (See Section BJ.5.06 for a description of the Process Concealed Stack.)
2. Obtains the processor index number (0-7) from the Processor Data Block. (See Section BK.1.02 for a description of the Processor Data Block.)
3. Executes a "clear associative memory" instruction in order to clear the associative memory of the processor on which it is executing.
4. Clears the connect flag array entry for the processor on which it is executing.
5. Restores the processor state to return control to the point at which the fault occurred.

Implementation Example.

The following machine code illustrates the initial implementation of the connect fault handler. In the example, pds represents the segment number for the Process Data Segment (Section BJ.5.04), fault_int represents the segment number of the Fault Interceptor Module, pseg represents the segment number of the Processor Data Segment (Section BK.1.01), pct represents the segment number of the Processor Communication Table (Section BK.1.04), and flag stands for the base location of the connect flag array.

```
rem      The following two instructions appear
rem      in the processor fault vector.
```

```
inhib  on
scu     =its(pds,4,*),* control unit into Concealed Stack
tra     =its(fault_int,connect),* go to connect handler
```

```
connect: sta      =its(pds,2,*) ,* save A temporarily
          lda      =its(pseg,proc_index) ,* get processor index no.
          rem
          rem      The assumed format of proc_index is zero   2*n,n
          rem
          cam
          stz      =its(pct,flag,a1) ,* send response
          lda      =its(pds,4,*) ,* reload A
          rcu      =its(pds,4,*) ,* restore control unit
          inhib    off
```