Published: 11/10/67

Identification

Gate_init
R. L. Rappaport

Purpose

Gate_init is a master mode subroutine involved in process initialization. This procedure enables a new process to cross from the hardcore ring to the administrative ring without leaving a history of the crossing. In this way, the process appears to have begun life in the administrative ring.

Discussion

A call to gate_init is the last thing done by procedure init_proc. The call is made after the new process has initialized the dynamic linking mechanism in its address space. Gate_init is called on fault_stack. The purpose of gate_init is to switch rings, to switch stacks and to call the procedure charged with initializing the administrative ring in the new process. The above is accomplished in a straightforward way.

Upon entry, gate_init initializes the Gatekeeper's data base to show that the new process is in the administrative ring and furthermore, that it originated in this ring. Then a call is made to subroutine ring$load (see BD.3.05) to effect the transition to the administrative ring. Since fault_stack is accessible in master mode in this ring, no difficulties are encountered because of the switch.

Once in the new ring, we switch to the normal paged stack in this ring. Since gate_init is never returned to the call history on fault_stack is discarded. That is, fault_stack is reset to appear empty.

At this point gate_init is ready to transfer control to subroutine init_admin (see BJ.9.04) which will continue process initialization. However, gate_init is a segment that was pre-linked at system initialization time when init_admin was not available. Therefore, the call must be made indirectly through the process definitions segment (pdf, see BJ.1.06). That is, gate_init makes the call:

```
     call pdf$init_admin,*
```

in order to give up control.