TO: MSPM Distribution
FROM: J. H. Saltzer
SUBJECT: BL.9.02
DATE: 03/06/68

The attached revision of BL.9.02 reflects a major revision in the organization of the Interrupt Interceptor and the Interrupt Initializer. The Interrupt Initializer now is much simpler: it sets several ITS pairs in the Interrupt Interceptor, and it overlays certain ITS pairs in segment "fault_vector".

Published: 03/06/68
(Supersedes: BL.9.02, 05/10/67)

Identification

Interrupt Initializer
L. J. Lambert, P. Schicker, J. H. Saltzer

Purpose

The Interrupt Initializer initializes the Interrupt Interceptor module: and the ITS pairs associated with the interrupt vector. This section assumes knowledge of Multics interrupt handling, (BK) and Multics Initialization (BL).

Introduction

When Multics initialization passes control to the interrupt initializer, the hardcore supervisor is in the following state.

1. All of the segments of the hard core supervisor which are required to initialize the interrupt interceptor have been loaded into core by the segment loader and all external segment references have been pre-linked. (All of the data bases used for interrupt handling have been provided, although their contents may be zero.)

2. The system communication segment (BK.4.01), which describes the hardware available to the Multics system has been initialized.

Initialization of the Interrupt Interceptor

The Interrupt Initializer must set the ITS-pointers internal to the Interrupt Interceptor that are used to store and reload registers before the linkage is established and after the base registers have been reloaded.

Since the Interrupt Interceptor must establish its own linkage after every interrupt, an initialized set of base register settings are stored within the Interrupt Interceptor. The standard pairings and the correct values for lp_lb are set by the Interrupt Initializer.

Interrupt Vector Initialization

Initialization of the interrupt vector involves storing ITS-pointers which point to entry points in the Interrupt Interceptor in the segment "fault_vector". These ITS-pointers overwrite the ITS-pointers provided placed there at the beginning of initialization. As soon as these pointers are overwritten, the Interrupt Interceptor becomes "live"; that is, any future interrupts will be handled by it.

The calling sequence to the interrupt initializer is:

```
call iiinit;
```