TO: MSPM Distribution
FROM: W. R. Strickler
SUBJECT: BX.15.06
DATE: 04/10/68

The system operator command, startup, has been revised to include a standardized method of relaying error information from the System Control process to the System Operator.

Published: 04/10/68
(Supersedes: BX.15.06, 12/29/67)

Identification

System Operator Command to Create New Processes
startup
W. R. Strickler

Purpose

Among the processes listed as system processes in BQ.1.02, only some in the System Control Process-Group exist after Multics is initialized and the system operator's process-group is created. Certain of the other processes in the list are created as the result of the system operator issuing the startup command. In Initial Multics initialization is required for the backup system.

Usage

The system operator types

```
     startup process_name
```

where process_name is the name of a system process or process-group which the system operator is authorized to create.

Implementation

The contents of data segments "operator_comm" and "request_name" (that is, the structures op_comm and op_req) are described in BX.15.05. The data segment "startup" in the System Control request directory contains the structure:

```
     dcl 1 process_name based (sp),

         2 name char (24),        /* of process to be created */

         2 process_id bit (36),   /* of created process */

         2 state fixed bin (17),  /* status of completed request */

         2 info char (64);        /* error description */
```

The startup command takes the following steps:

1. Place the argument process_name in sp→process_name.name.

2. Create an event channel over which System Control can reflect completion of the startup request; place the name of the event channel in rp→op_req.ref_chn.

3. Place the name "startup" in rp→op_req.req_name.

4. Signal System Control, over the channel named in p→op_comm.op_req_chn, the event whose ID is in rp→op_req.ref_chn.

5. Call the Wait Coordinator (BQ.6.06) to wait for the reflection signal from System Control.

6. Examine status returned by System Control. If sp→process_name.state is not zero, inform the operator of the nature of the error by calling write_out (BY.4.02) with the argument sp→process_name.info.

7. Delete the event channel created in step 2.

8. Return.