TO: MSPM Distribution
FROM: M.R. Thompson
SUBJECT: MSPM Sections BG.3.00 - BG.3.05
DATE: 05/26/67

The MSPM sections BG.3.00 - BG.3.05 replace section BG.3
and reflect a reorganization of the Segment Control Module.
In general the functions of Segment Control are unchanged
with the following two exceptions: some of the information
retrieving primitives such as getsegno, getname, segstatus,
setseglock, newmode, and oldmode no longer exist as these
functions have been moved to the Segment Management Module
(BD.3), and a new set of primitives have been added, the
Ring Simulation Module, to handle ring changing and validation
functions. Otherwise the only changes are implementation
details, some primitive names and arguments.

Identification

Introduction to Segment Control
R. C. Daley, M. R. Thompson

Purpose

Segment control maintains records of all segments known
to the current process and makes these segments available
to the process upon request. Information concerning these
segments is maintained in the known segment table (KST)
which is manipulated exclusively by segment control.
Segment control also provides information concerning active
segments in the active segment table (AST) and the descriptor
segment table (DST) for the exclusive use of segment control
and page control. In addition, segment control maintains
an entry in the process segment table (PST) for each process
which is currently active (see BJ.9.01).

Introduction

Segment control is divided into five separate modules,
the System Interface Module, the User Interface Module,
the Process Load Module, the Segment Utility Module, and
the Ring Register Simulation Module. The following list
gives a brief description of the functions of these modules.

1. The system interface module - This module contains
    primitives which are privileged and provided for
    the use of other supervisory procedures and the
    backup and multilevel systems (see Section BG.3.01
    of this manual).

2. The user interface module - This module contains
    all of the unprivileged primitives of segment
    control. These primitives provide various service
    functions for the user relating to segments already
    known to his process. The user interface module
    is fully specified in section BG.3.02 of this
    manual.

3.   The process load module - This module provides
        primitives for the use of various modules of the
        hard-core supervisor. These primitives are
        specifically designed to assist in the loading
        and unloading of processes. The process load
        module is fully specified in section BG.3.03 of
        this manual.

4.   The segment utility module - This module consists
        of several primitives which are provided for the
        exclusive use of segment control procedures. These
        primitives are provided primarily to assist in the
        manipulation of the AST and KST. The segment
        utility module is fully specified in section
        BG.3.04 of this manual.

5.   The ring register simulation module - This module
        is separated from the others only to minimize
        the changes in the file system when and if a ring
        register is added to the 645. It will go away if
        this hardware becomes available. The ring register
        simulation module is specified in section BG.3.05
        of this manual.

Errors

In all of the primitives of segment control, the last
argument in the calling sequence is an error code. If
a called procedure detects no errors in its work, this
argument will be set to zero on return; if an error is
detected, it will contain an error number which encodes
the type of error and the procedure which detected it.
It is the responsibility of the caller to test the error
code and take the appropriate action.

A list of all the error codes used by the basic file
system is contained in section BG.14.