Published: 02/09/68
(Supersedes: BL.7.03, 04/05/67)

Identification

Data Base Initializer - Initialization datmk_
dbi
D. H. Johnson, N. I. Morris

Purpose

This section describes the data segment growing procedure that operates during Multics initialization and process creation. It is used in the implementation of PL/I static storage to grow storage regions as needed. It is called by the initialization pre-linker (see BL.7.01 and BL.7.02) when it recognizes a "trap before link" request while attempting to set a link pair. dbi has the same purpose as datmk_, the data segment grower procedure which operates under Multics (MSPM BP.4.01). It replaces the Multics precedure during Multics initialization and process creation so that the special requirements imposed by these operating environments may be properly handled. dbi differs from datmk_ in that it does not call the SMM, generate_ptr, or link_change$make_definition. Since it is used at pre-linking time, all storage regions are grown immediately instead of when first referenced.

Usage

dbi is called only by force_link in the pre-linker with the following calling sequence:

```
call dbi$datmk_ (link_pair_ptr, arg_list_ptr,
     table_manager_ptr);
```

link_pair_ptr is a pointer to the linkage pair in the linkage section of the procedure being pre-linked.

arg_list_ptr is a pointer to the procedure's argument list (see Section BP.4.01).

table_manager_ptr is a pointer to a procedure which may be called with a segment name in order to obtain pointers to that segment and its associated linkage segment. In Multics initialization, the table manager will be slt_manager$get_text_link_ptr (see Section BL.2.02); in process creation it will be part of the process initialization table manager (see Section BJ.9.07). The calling sequence of the table manager is as follows:

```
call table_manager (segment_name, text_ptr,
     link_ptr, err_sw);
```

Execution

1. Force_link is called recursively to attempt to make a link to the data region specified by link_pair_ptr.

2. If force_link was successful, the data region already exists. Return.

3. Otherwise, the table manager is called to get pointers to the text and linkage segments. If either text or linkage are not found, dbi returns without growing the data base.

4. The new definition is added to the linkage segment and the segment is grown by the specified number of words. That is, word 0 of the text is incremented by the specified number of words (see BP.4.01).

5. If the argument list in the EPL-compiled code did not specify an initialization precedure, return.

6. force_link is called again to make the link to the new data region. If it was unsuccessful, dbi will simply return.

7. The initialization procedure specified in the EPL-compiled code is called.

8. Return.

Restrictions

1. dbi may not be used to grow data regions in segments with combined linkage sections. If this inadvertently happens, chaos will result.

2. The first word of any segment to be grown with dbi must be reserved for a count of the number of locations used in that segment. If the count is initially zero, the first data region will be grown at location 2.

3. Both the text segment to contain the data region and the linkage segment to contain the data region's linkage definition must exist at the time dbi is called.

Note

In Multics initialization and process creation, trap-before-linking references to datmk_ will actually result in a call to dbi. This is accomplished by making the segment name of dbi be datmk_ in these environments.