

Published: 11/17/66

Identification

Data segment grower  
 datmk\_  
 D. B. Wagner and M. D. McIlroy

Purpose

Datmk\_ is used in the implementation of PL/I static storage to "grow" storage regions as needed. Normally it is called only through an out-reference in a linkage section which specifies it in the "call-before-linking" option.

Usage

Use of datmk\_ is specified in EPLBSA by:

```

          segref    datmk_,datmk_
          segref    segment,symbol(datmk_(arglist))
          ...
arglist  dec       size
          dec       initialswitch
          arg       initializer
  
```

Here segment and symbol are the names of a segment and an in-reference in that segment's linkage section. At execution time, the first reference to symbol, e.g. the instruction

```

          eapbp    symbol
  
```

causes a trap to the linker, which in turn calls datmk\_. If segment is not active in the process, datmk\_ creates it and its linkage section. Then if symbol is not listed as an in-reference in segment's linkage section, datmk\_ grows segment by size words and creates the in-reference pointing to the newly-grown storage.

If initialswitch is non-zero, datmk\_ fills in the faulting link pair and calls the user's initializing procedure located at initializer. This call has the form of a call to a PL/I internal procedure (see BP.3.00 for details)

with no arguments. Since this call does not go through the linkage section, if the initializing routine uses the base pair  $lb \leftarrow lp$  it must obtain the proper values itself. Assuming that  $lb \leftarrow lp$  is properly set, however, the initializing routine may freely refer to symbol.

Finally `datmk_` returns to the linker, which uses the RCU instruction to restart the user's program at the faulting instruction. By the time this instruction has finished executing, the data region has been grown and initialized, and the instruction has had its proper effect.

#### Implementation

`Datmk_` is called by the linker as follows:

```
call datmk_ (argpointer,panelpointer);
```

where argpointer is a pointer to the user's argument list specified in the segref pseudo-op, and panel is a pointer to stored machine conditions as follows:

words 0-7	SCU information
8-15	base registers
16-23	arithmetic registers