



Massachusetts Institute of Technology
Information Processing Center
Cambridge, Massachusetts 02139

Saltzman
file - redesign
purposes

MEMORANDUM

To: Hardcore Group
From: C. Franklin, T. VanVleck, R. C. Daley, and R. Freiburghouse
Subj.: The STACK and STORAGE Conditions
Date: 17 December 1971

It would be highly desirable to be able to recover from out-of bounds errors on the stack. Currently such errors cause process termination, and there is no easy way to find the cause. This memo proposes a way to implement a new system condition, called STACK, and discusses the ANSI Standard PL/I condition STORAGE.

Implementation

The STACK condition can be implemented by modifying four modules as follows:

1. `makestack` should make the stack boundary 62 pages instead of 64 pages
2. `boundfault` should detect if the faulting segment is the stack (by looking at `pdf$stacks`), and, if so, it should append two more pages onto the stack boundary and raise the STACK condition (i. e., return new error code to the FIM).
3. `error_table` should have the STACK error code added to it.
4. The `default_error_handler` should raise the ANSI standard condition STORAGE when the STACK condition reaches it.

The reason for having two conditions, STACK and STORAGE, is that the STORAGE condition can also be raised for reasons other than stack overflow, and it is better to have two conditions than one condition whose subcases are distinguished by various values of the ONCODE built-in function.

Justification

1. Highly desirable to catch out-of-bounds on stack and prevent process termination.
2. Appears easy to do.
3. Implements more of ANSI PL/I while also helping the non-PL/I user.