

Published: 07/18/67

Identification

Perform an abnormal return  
unwinder  
M. A. Padlipsky

Purpose

An "abnormal return" is an exiting from a procedure to a point other than the one (immediately after the call) which would normally be returned to through use of the standard Multics return macro. The target point is a PL/I-type label, which contains a Stack pointer that corresponds to a frame which has not yet been returned to in the process at hand; typically, this label is passed as an argument. Because of the fact that the locus of control of a process may contain protection-ring crossings, the abnormal return must be effected by a procedure which is part of the Multics protection mechanism. Therefore, to perform an abnormal return to label lbl, the user calls procedure unwinder with lbl as argument. Labels at which inter-ring abnormal returns are acceptable must be specifically established as "doors" in the protection walls by use of the setacl command (see BX.8). Detailed discussion of the Unwinder is to be found in BD.9.05. Definitions of terms are to be found in BD.9.00.

In performing an abnormal return, the Unwinder fulfills three basic functions. The first two are essentially system-oriented: balancing the books of the protection mechanism, and releasing Stack frames of procedures which are being bypassed (i.e., to which neither normal returns nor abnormal returns will be taken). The third function of the Unwinder is to offer the user the ability to perform "unfinished business" in the bypassed procedures. That is, starting with the procedure which calls it, the Unwinder will invoke any procedures which are found as handlers for the condition name "cleanup", as deposited in the signal vector by calls to condition; see also BD.9.04, BY.12.04. For example, a "cleanup" handler might free storage which was allocated by the procedure it is established for. On normal return from a procedure, "cleanup" handlers should be removed from the signal vector by calls to reversion (BD.9.04, BY.12.05) so that the Unwinder will not erroneously attempt to invoke them in the event of a subsequent abnormal return.

### Usage

The calling sequence is

```
call unwinder (lbl);
```

with declaration

```
dcl lbl label;
```

where

lbl is the target of the abnormal return; i.e., the label to which the present procedure is to exit.

### Error Handling

The Unwinder employs the standard Multics error-handling technique (see BY.11). There are three errors which could be of interest to the user: 1) The inter-ring target label is not a door; 2) The inter-ring target label is a door, but does not correspond to a Stack frame of the process at hand; 3) The intra-ring target label (which need not be a door) does not correspond to a Stack frame of the process at hand. If the user has a condition handler established for "Unwinder\_err", it may return to the Unwinder, in which case the Unwinder returns to its own caller.

It is most important to note that the target label must belong to a Stack frame (that is, be in a procedure) which has not been returned to at the point in the process when the abnormal return is taken; particular care must be taken in this respect when the label is not passed as an argument, but is, for example, treated as an "external static" entry in the PL/I sense.

### Implementation

Details of the implementation of the Unwinder are to be found in BD.9.05.