Published: 07/19/67

Identification

Inspect condition-handler push-down list
find_condition
M. A. Padlipsky

Purpose

It is sometimes useful to be able to examine the contents of the push-down list of handlers for a given condition (see condition, BY.12.04). Procedure find_condition is furnished for this purpose; it operates in the protection ring it is invoked from.

Usage

The calling sequence is

```
call find_condition (condname, n, proc, flag);
```

with declarations

```
dcl condname char(*), (n, flag) fixed bin(17), proc label;
```

where

condname is the name of the condition whose handler list we wish to examine.

n is the number of places down in the list to examine (n = 0 indicates the top of the list)

proc (returned by find_condition) is the handler indicated in the nth entry in the list for condname, or is the last entry in the list if there are less than n + 1 entries.

flag is set either to 1) zero, if proc is indeed the nth entry, or 2) m, where m is the number of places down in the list proc is, if proc is the last entry rather than the nth, or 3) -1, if there is no list.

Implementation

1. Call get_ring_no (BY.12.07) to determine r, the current ring number.

2. Call generate ptr (BY.13.02) for condname in <signals_r>. If the pointer returned is null, condname has not been established; set flag = -1 and return.

3. Set up for n + 1 iterations.

4. Inspect the back pointer in the current entry for condname. If it is 0, proceed to step 6.

5. If n + 1 has not been searched, make the back pointer the current pointer and return to step 4. If n + 1 has been reached, set flag to 0, set the entry data portion of the current entry into proc, and return.

6. There are not n + 1 handlers in the list; set flag equal to the current value of the loop index minus 1, put entry into proc and return.