Identification

Calling a Procedure Whose Name is Not Explicitly Known

fake_call

W. R. Strickler

Purpose

It is sometimes necessary for a procedure to call or obtain a pointer to another procedure whose name is not known until the calling procedure is executing; for example, the name of the called procedure could be obtained from a table.

The procedure fake_call has as arguments the character string representations of the procedure and entry (if any) and fabricates a call to the procedure (§entry). The called procedure cannot have arguments.

The entry fake_call§ptr returns a pointer to the procedure (§ entry).

Usage

Either

    call fake_call (name, entry);

or

    call fake_call§ptr (name, entry, p);

The arguments name and entry are character strings (either varying or non-varying); the argument p is a pointer. A call to fake_call results in a call to an entry y in a procedure x; the entry fake_call§ptr returns a pointer to x§y, determined as follows:

    If entry is not null, x§y = name§entry

    If entry is null and the name string contains the "§" character (name = "alpha§beta"), then x§y = alpha§beta.

    If entry is null and name does not contain the "§" character, then x§y = name§name.
Implementation

The arguments name and entry are "converted" to adjustable non-varying strings (see BY.10.03) seg and sym by:

```
call cv_stringsecs (name, seg);
call cv_stringsecs (entry, sym);
```

If entry is null, then the index function is used to determine the location, if any, of the "f" character in the string name. If the character is present, the appropriate substrings of name are converted into seg and sym; otherwise, name is converted into both seg and sym:

```
dcl li fixed bin (17);
li = index (seg, "f");
call cv_stringsecs (seg, sym, li+1);
if li = 0 then call cv_stringsecs (seg, seg, 1, li-1);
```

Call generate_ptr$initiate (see BY.13.02) to get a ptr to seg|sym; this pointer is used in building the 216-bit string which is the argument of fake_entry$call (see BY.10.01), which forces a call to seg|sym. The following code is used to invoke generate_ptr:

```
dcl class fixed bin (17);
dcl 1 lb,
   2 (pt, sp, ex) ptr;
call generate_ptr$initiate (seg, sym, lb.pt, class, 0);
```

If the ptr entry was called, lb.pt is assigned to p, and the procedure returns.

Otherwise, the following code is executed to call x$y:

```
dcl b bit (216) based (ep);
ep = addr (lb);
lb.sp = null; lb.ex = null;
call fake_entry$call (ep->b);
return;
```