Published: 06/01/67

Identification

The EPL run-time routine, index_
index_$indexc_,
index_$indexb_

Ruth A. Weiss

Purpose

Index_ implements the PL/I function index where the function value=integer=

a) the index of the first element of the first argument such that starting at this element, the second argument appears as a substring
b) 0, if no such argument satisfying (a) exists or if either of the arguments is of length 0.

For indexc_, an element is 9 bits.
For indexb_, an element is 1 bit.

Usage

The two possible calls are:

```
call index_$indexc_ (c1,c2,i)
call index_$indexb_ (b1,b2,i)
```

c1 and c2 are character strings, varying or non-varying
b1 and b2 are bit strings, varying or non-varying
i is a binary integer

The statement

```
i = index (a,b)
```

is implemented in EPL by one of the following calls:

```
call  stgop_$ixcs_(a,b,i);

call  stgop_$ixbs_(a,b,i);
```

(See BN.7.09 for a description of stgop_$ixcs_ and stgop_$ixbs_.)

Implementation

There are two separate substring search loops:

1. For substrings of < 36 bits.
   Here only part of a word is compared each time. The machine operation CMK is used to avoid loading and shifting for every element. Because of the overlap this loop is much slower than 2.

2. For substrings of ≥ 36 bits.
   A search is made for a match on the first 36 bits of the substring. The accumulator and quotient registers are fully utilized for the comparisons. When a match is found, if the substring = 36 bits, the match is complete. If the substring > 36 bits, a dummy specifier is written for the first argument of the substring. Then strcmp_$eqb_ (See BN.7.12 for strcmp_) is called with these 2 arguments. If the strings are equal, the match is complete. If they are not equal, the search in the loop is resumed.

Errors

If either of the arguments is not a string, stops with oct 0.