Published: 10/11/66

Identification

"Left-hand" expression evaluator

setvalue

D. B. Wagner

Purpose

The probe command must accept requests of the form

```
     set variable = expression
```

where expression is an ordinary debugging expression and variable is a variable name, a subscripted variable, or a "pseudo-variable" (in the sense used in the PL/I manual) as in the request

```
     set c(alpha$7) = 0
```

or even worse,

```
     set substr(unspec(a(7)),7,1) = "0"b
```

The procedure setvalue is the complement of the procedure evaluate described in BY.6.04. It takes a tree-structured representation of an expression (from the left-hand-side of an assignment) and sets the contents of the data referred to by it to a specified value, making whatever data-type conversions are necessary. Normally the specified value has been obtained previously through a call to evaluate for an expression on the right-hand-side of an assignment.

Usage

The call is

```
     call setvalue (tree_pointer, data_pointer, node_pointer);
```

The declaration associated with the arguments is:

```
     dcl (tree_pointer, data_pointer, node_pointer)ptr;
```

Tree_pointer points to an operator-operand tree representation (created by parse: see BY.6.01) of an expression which might have appeared on the left-hand-side of an assignment.

Data_pointer and node_pointer give a value previously produced by evaluate. (See BY.6.04: Data_pointer points to raw data and node_pointer points to a genuine or imitation symbol table entry which gives information as to how the data is to be interpreted.)

Setvalue scans the tree, calling evaluate to evaluate subscripts etc., and sets the value of the data referred to the value given by data_pointer and node_pointer.

Eventually setvalue will recognize all of the functions called "pseudo-variables" in the PL/I manual (these are the functions allowed on the left-hand-side of an assignment statement) and the c and cr ("contents" and "contents of register") functions of the debugging language, and will handle subscripted variables properly. The initial implementation will not handle subscripting, and will handle only the substr, c, and cr functions.