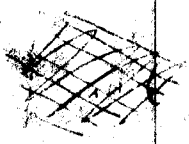
read xwtlg

Syntax of parameters
(no blanks)

objloc ::= objprim | objprim : id
objprim : id : objprim |
objprim * octexp | * octexp

objprim ::= id | (objloc)



wordloc ::= objprim * octexp | * octexp

octexp ::= octterm | octexp + octterm |
octexp - octterm | - octterm

octterm ::= octprim | octterm * octprim

octprim ::= id | octinteger | (octexp)

wordexp ::= wordpart | wordexp , wordpart

wordpart ::= decimalinteger : octprim

---

id may contain letters, digits, .

Semantics

Each expression has a value. This value is computed by first computing the value of each subexpression. The rule used in forming the main expression then determines a function to be applied to the values of the subexpressions to determine the value of the main expression.

For each rule we will define the function determined by it. For some rules with exactly one subexpression that function is the identity function. Those rules will not be listed.

There are several different kinds of values, they are:

datum	a word of less than or equal to 60 bits
object	a capability
object locations	a clist and an index within it the users subprocess full clist and an index within it a directory, name and access key a search list and name

data locations	a file and index within it
	Theseus subprocess full core and address within it
(?)	Theseus choose pointer variables within it

a name	ascii text
a variable	can contain either object or datum

certain primitive expressions have values as follows

rd	a name
octal integer	datum
decimal integer	datum

The locations are represented ~~as~~ in an obvious way, illustrated by

CLISTSLOT
~~CONST~~ (~~CONST~~ C, I)

meaning C must be a clist and I an integer and the whole expression means the Ith slot within clist C.

The functions determined by rules which are not robot. by functions are as follows:

~~objloc ::= objprim : id~~

~~The value of objprim must be a record~~

~~The value of this rule~~

We state the rule

The notation is to 1st state the rule. Then we tell how to compute the desired function. The name of a subexpression stands for the value of the subexpression. val(exp) means either the value of exp if it is an object or datum or the contents of the location if it is a datum loc or objectloc.

objloc ::= objprim : id

~~val(objprim) must be a record~~

~~Error~~ secval(val(objprim), id)

objloc ::= objprim : id : objprim

directory(val(objprim$_1$), id, val(objprim$_2$))

objloc ::= objprim * outexp

clistslot (val(objprim), outexp)

objloc ::= [illegible] * outexp

userfullclist (outexp + currentuser clistbase)

↑ needs explanation

objprim ::= id

if there is a variable of name id then that variable
else srcnlist (currentscanlist, id)

wordloc ::= objprim * outexp

filebaddress (val(objprim), outexp)

wordloc ::= * outexp

userfullcore (outexp + currentuser corebase)

↑ needs explanation

octexp ::= octterm

use obvious arithmetic operation on a full 60-bit word

octprim ::= id

if id is the name of a variable whose value is a datum,
then that datum,

wordpart ::= decimalinteger : octprim

a pair <decimalinteger, octprim>

wordexp ::= wordpart

2nd half of wordpart

wordexp ::= wordexp , wordpart

wordexp ← $2^{1\text{sthalf(wordpart)}}$ + 2ndhalf(wordpart)

(as unsigned 60-bit integer)