**2013-11-05**

Goal ==> <Pascal program>

<pascal program> ->

<header> Create Symbol Table

<declarations>

<begin-statement> Kill Symbol Table

.

<header> ==> {empty}

<declarations> ==> {empty}

<var decl><declarations>

<const decl><declarations>

<label decl><declarations>

<procedure decl><declarations>

--------------------------------------------------------------------------------------------------------

Var Declaration

<var decl> -> [ var <namelist> : <type>; ]+

in declaration we expect new identifiers, thus not present in the symbol table.

In statements we expect identifiers that are known, thus present in the symbol table.

{1}[i][TK\_VAR][TYPE][address]

1. Parse <namelist>, create linked list of names
2. Ensure that all names are not in the top symbol table or in the name list.
3. Match (TK\_COLON)
4. parse type create records for all names on the list
5. allocate
6. Delete list

How to compute the address:

size = size(type)

Integer 4

Character 1

Real 4

dp is allocation ptr in data,

intially 0

addr <- dp

dp <- dp + size

-----------------------------------------------Align memory --------------------------------------

addr <- align( dp, size ) ((dp / size) == 0) ? dp : (round Up(dp / size) \* size)

dp <- addr + size;

---------------------------------------------------------------------------------------------------------------------------------

Body of the program

=================================================================================

<begin-statement> -> begin <statements> end

<statement> -> <begin statement>

<if statement>

<while statement>

..........................

<assignment> TK\_VAR

<write statement>

we know it's assignment if we see variable already in the symbol table

<assignment> => <LHS> := <RHS>

LHS -> TK\_VAR

RHS -> E

//Save info about LHS

(tLHS,Addr(LHS)) //get address of left hand side

match(TK\_ASSIGN)

t(RHS) = E();

using our example:

push 0;

push 4;

add

cvr

pop 8;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LHS/RHS | I | R | B | C |
| I |  | Cvi | X | X |
| R | cvr |  | X | X |
| B | X | X |  | X |
| C | X | X | X |  |

--------------------------------------------------------------------------------------------------------

Example Program

var i,j : integer;

r:real;

begin

r = i + j;

end.

==================================================================================

Comparison Expression.

L -> E | E=E | E<E | E<=E | E>=E | E<>E | E>E

Except for one place:

F -> (E)

this now becomes:

F -> (L)

E -> E'

E' -> <E | =E | <=E | >=E | <>E | >E

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| < | I | R | B | C |
| I | Lss/B | Conversion | X | X |
| R | Conversion | Flss/B | X | X |
| B | X | X |  | X |
| C | X | X | X |  |