Today: - var decls
- caveat

```
ST_DR st_lookup(SID id,
                int *block);
int junk;
dr = st_lookup(id, &junk);
```

```
var
  x,y,z : Real;
  a,b,c : Integer;
  f,g : array[2..6] of Real;
  p : CharPtr;
  q : ^Integer;
```

Install each var in the symbol table as a GDECL

enroll - st_enter_id
install - st_install
requires ST_DR

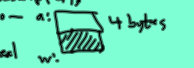
2. Emit assembly code:
assembler directives that:
1. Declare the var as global
2. Emit a label for the var
3. Align var properly
4. Allocate enough space for the var

Two backend routines:
be_global_decl(char *, unsigned size, int align)

Static Storage;
x,y,z : Real;

be_global_decl("z", 8, 8);
be_skip(8);

a : Integer;
be_global_decl("a", 4, 4);
be_skip(4);



w: Real
w:

be_global_decl("w", 8, 8)

compute size & alignment requirements for any given TYPE

size rules:

Integer	— int	4
Char	— unsigned char	1
Boolean	— char	1
Single	— float	4
Real	— double	8
pointer to anything	— * —	4

1..8 same as the base type (in this ex. Integer)

```
Var
  x : 1..5;
```

array[low..high] of elementType
size is the size of the elementType, times the number of allowed indices (high - low + 1).

Multi-dim array:
array[l1..h1, l2..h2, ..., lk..hk] of ET
size is size of ET
x (h1-l1+1) x (h2-l2+1) x ... x (hk-lk+1)

array[3..5] of ^function : Integer

array[3..7, 1..9, 0..9] of ^Integer
5 4 10 4
size = 800

Alignment: for all basic data types (incl. pointers), alignment=size
arrays: alignment is same as element's
proc's & funcs have no well-defined alignment or size

Mar 15-11:02 AM

Getting the indices & element type of an array:

TYPETAG
ty_query(TYPE)
what kind of type?
TYSIGNEDINT — Integer
TYDOUBLE — Real
TYPTR — ^ ...
:
TYARRAY

TYPE
ty_query_array(TYPE, INDEXLIST *)
element type
TYPE array type
INDEXLIST output param

:
VAR
p,q : ^Integer;

can call for all var decls → { resolve pointer types; do the installs }

forward refs are not allowed in the VAR decl section.

variable_declaration:
id_list ':' typedenoter semi
{ resolve "all" ptr-types;
process the installation & assembly code for each id;
}
;

Integer, Real, etc. are all pre-installed in the symbol table as TYPENAME's.

Integer — ty_buildbasic(TYSIGNEDINT)
Real — ty_buildbasic(TYDOUBLE)
:

Mar 15-11:57 AM