Data definition directives

* asm must obey the rules of high level programming

data definition = declarations (attributes specification) + allocation (resv. required mem. space)

Unique! — it is NOT unique (ex: int, float...) — Unique

Q: by whom is the allocation?

most of the time, by the assembler, but not all variables are allocated by the compiler (aka global variables in the global data segment)

some are allocated at runtime (local variables are allocated on the stack only addressable at runtime)

! diff. between allocating and initializing

every programming language has 3 logical spaces:
- global data segment (GDS)
- stack (local variables)
- heap (dynamic variables)

not allocated at runtime but during it

ex: p = malloc(how_many_bytes)
alloc()

how you ask for memory space

returns a pointer to that mem. loc.

! a dynamic variable doesn't have a name
↳ they work with pointers bcs. they don't have a name, only a mem. area
↳ accessible only by means of associated pointer

Structural

Variable = (name, set_of_attributes, reference (address), value)

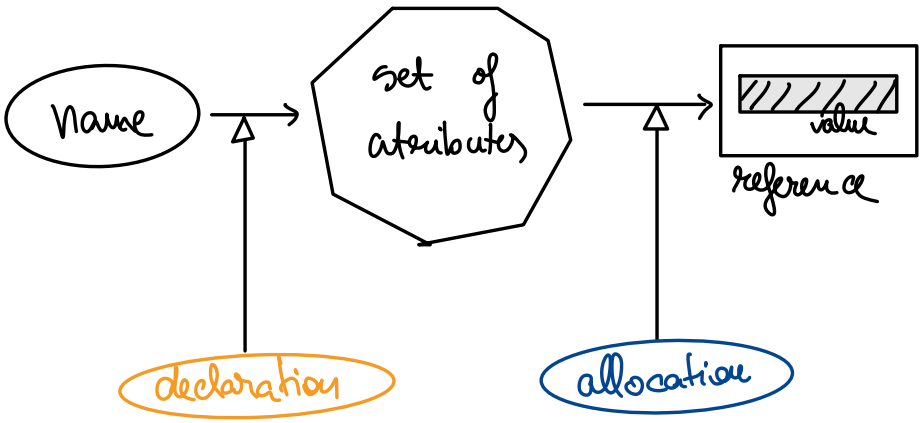
- data type — size, representation, associated operations
- scope (domeniu de vizibilitate) → portion of code in which that variable is visible (spacial visibility domain)
- extent – lifetime (temporal)
- memory class (auto, register, static, extern) C / C++

ex: register int a;
↑ if possible, it is allocated in the process. register for fast access
ex: a loop, going into memory is 1000x slower than accessing a register

extern float f;
↑ it will be imported, but this is a declaration in the modules
allocation is unique, only done once, but declaration can be done mult. times

static int a;
↑ keep it local and stop the implicit exporting
↑ if it is a local => alloc. in GDS
→ lifetime = whole prog.

ex: float f1 (....) {
...
static int x = 17;
...
}
keeps its value between 2 calls



declaration → moment when the name is linked with the atributes

EQU directives

* see slides ! study at home

! dereferencing is only possible at runtime

a41 db a2+1 → syntax error
↳ an OFFSET cannot fit a BYTE

a41 dw a2+1
8 + 1 = 9 determinable at asm time
value at 9 bytes from start of memory