

Memory allocation

Static Allocation a.k.a stack allocation (fixed in size)
data structures that do not need to grow or shrink
such as global and local variables e.g. `char name[16];`

➡ done at compile time

✓ restricted, but simple and efficient

Dynamic Allocation a.k.a heap allocation (change in size)
data structure that might increase/decrease in size according to different demands e.g. `name = (char *) malloc(16);`

➡ done at run time

⦿ general, but difficult to implement **(our focus today)**

Heap allocation more concretely

- ➔ Manage contiguous range of logical addresses
 - `malloc(size)` returns a pointer to a block of memory of at least `size` bytes, or `NULL`
 - `free(ptr)` releases the previously- allocated block pointed to by `ptr`