CS61 Section 1

September 11, 2019

Memory + Syntax

Object

- region in memory that stores a value
- must occupy different pieces of memory

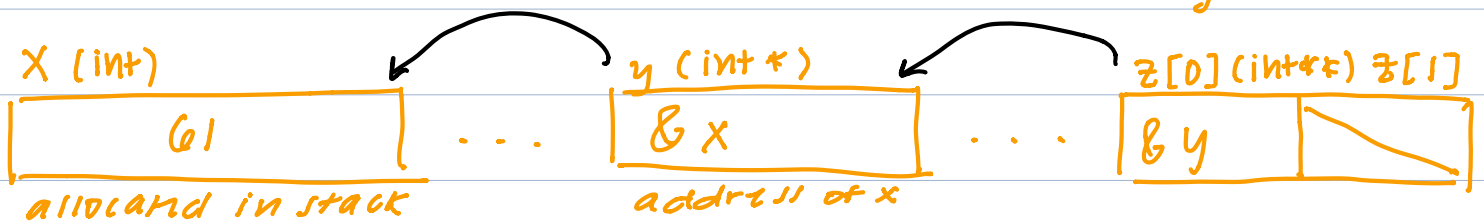
malloc returns void *

int * * z[2] = {&y, NULL}

z[0] = &y

* z[0] = y = &x

* &x ⟷ int* y = &x ; *y



*z ⟷ z[0] *(z+1) ⟷ z[1] } pointer arithmatic

* means de-reference and return value

& means to get the memory address (0x57...)

void * just doesn't have a type and needs to be cast

Which are objects?

x : yes

x+1 : no (value but not object)

&x : no (value of address but not object)

the string "x = %d\n" : yes (array of chars), even w/o assigning to a variable

cat : prints file

Types, sizes, and addresses

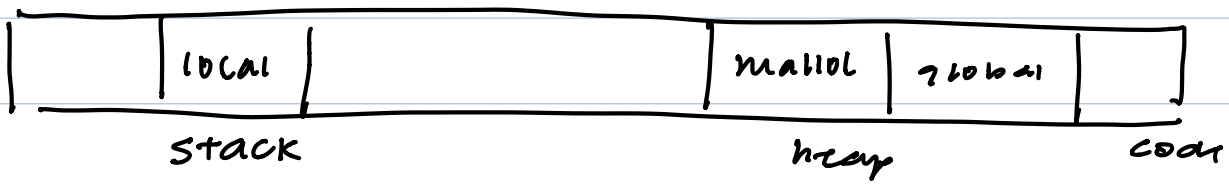
size/alignment depend on type

`vintptr_t` : for pointer arithmatic, the type of pointers
↳ 8 bytes of memory

`%p` : for printing out pointers

```
char* allocated_ch = (char*) malloc(sizeof(char));
```

malloced memory: in allocated_ch



C++ Standard Template Library

std::vector

- growable array

`.at(#)` ≡ `arr[#]` but more secure

`.push_back(#)` adds to back

`.back()` returns last

`.pop_back()` remove element at end

`.empty()` True if empty

`.clear()` erase

`.size()` returns # of elements

iterators

- more intelligent pointers, arithmatic works differently

`container.begin()` → iterator that points to beginning

`container.end()` → iterator that points to end

```
int my_array[4] = {...};

std::vector<int> my_vec = {...};

for(int* a = my_array; a != &my_array[4]; ++a)
    print(*a)
```
↳ pointer looping version

```
for(auto it = my_vec.begin(); it != my_vec.end(); ++it)
    print(*it)
```
↳ iterator looping

for each loop:

```
for(auto& a: my_vec)
    print(a)
```

- other methods
  - container.insert(iterator, value)
  - container.erase(iterator)
  - container.erase(first_iterator, second_iterator)

if it is an iterator, &* it gives the pointer to the element (except for v.end() which doesn't point to an element)

std::map<KEY, VALUE> * only way to do beam hitter portion of pset

std::map::find(#) returns an iterator

std::unordered_map<KEY, VALUE>

(hash table) — set online session 1 notes