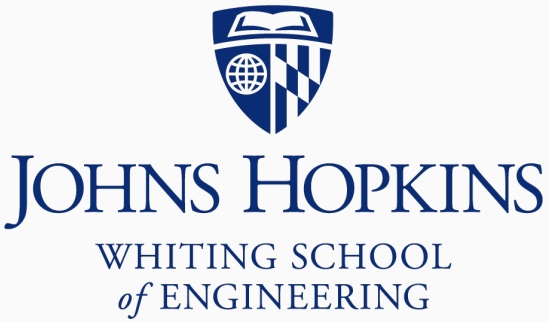
Memory

Ben Langmead

ben.langmead@gmail.com

www.langmead-lab.org

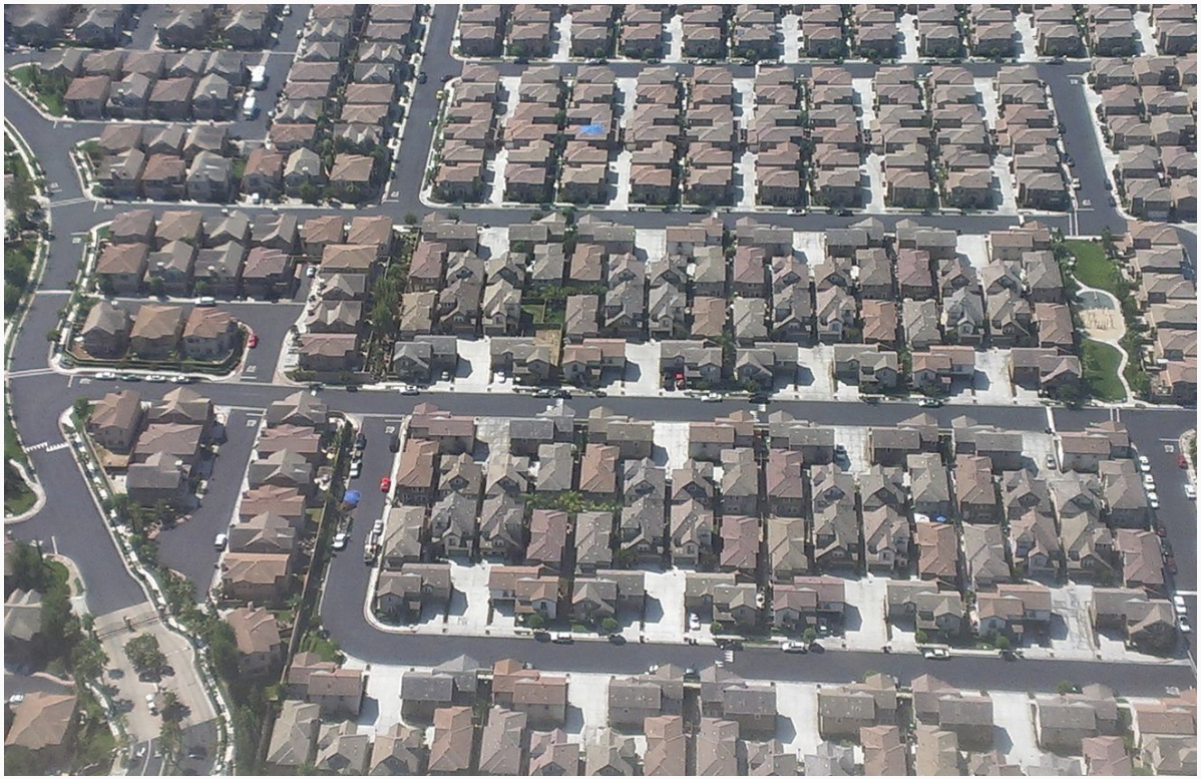


Source markdown available at github.com/BenLangmead/c-cpp-notes

Memory

Variables live in memory

Each variable has an “address” in memory, like a house address



commons.wikimedia.org/wiki/File:South-Los-Angeles-subdivision-houses-near-Darby-Park-Aerial-view-from-north-August-2014.jpg

Memory

Or like a post-office box:

Memory

In this example, `fahrenheit` and `celsius` are variables that live at addresses in memory

```c
#include <stdio.h>

int main() {
    int fahrenheit = 71;
    float celsius = 5.0 / 9.0 * (fahrenheit - 32);
    printf("%0.2f", celsius);
    return 0;
}
```

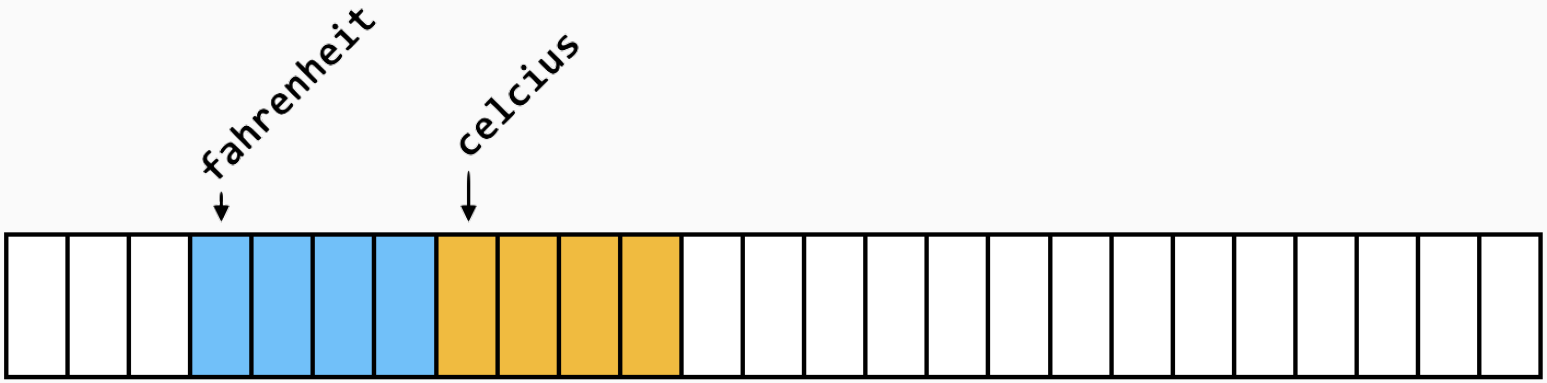
Memory

Memory is like a large array of bytes, like `char[]`

- An *address* is an *offset* into the array
- When a variable occupies >1 byte, address points to *first* byte

Memory

`int` and `float` both take 4 bytes, so we might picture them in memory like this:



We'll see diagrams like this frequently

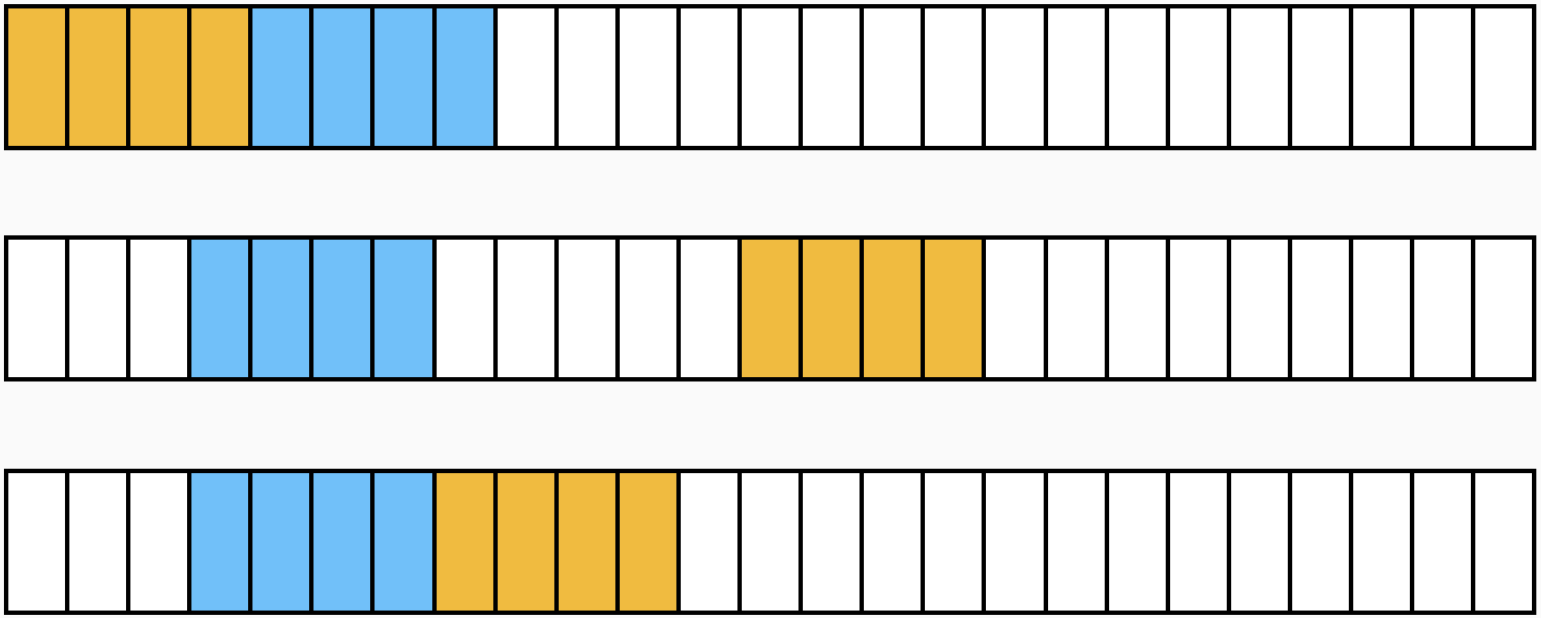
Imagine the leftmost slot's address is 0, the next is 1, etc

In this figure, `fahrenheit` is at address 3 and `celsius` us at 7

Memory

We can't generally predict what addresses our variables will be at

E.g. any of these would have been possible:



Memory

Putting & before a variable name *takes its address*

- We can print an address with `printf`:

```
#include <stdio.h>

int main() {
    int fahrenheit = 71;
    printf("fahrenheit lives at %p\n", (void*)&fahrenheit);
    //                                          ^
    //                                          address-of
    return 0;
}

$ gcc convert_fc_addr.c -Wall -Wextra -std=c99 -pedantic
$ ./a.out
fahrenheit lives at 0x7ffff24c142c
```

Memory

Two unfamiliar things here:

- The `(void*)` before `&fahrenheit`; we will talk more about this when we cover pointers (soon)
- The address itself: `0x7ffd56b72dcc`

`0x7ffd56b72dcc` is just a number!

- `0x` at beginning indicates it's base-16, or *hexadecimal*
- In base-16, digits `0` – `9` aren't enough so use a – `f` as well
- This number is 140,726,058,298,828 in decimal

Pointers

xkcd.com/138/