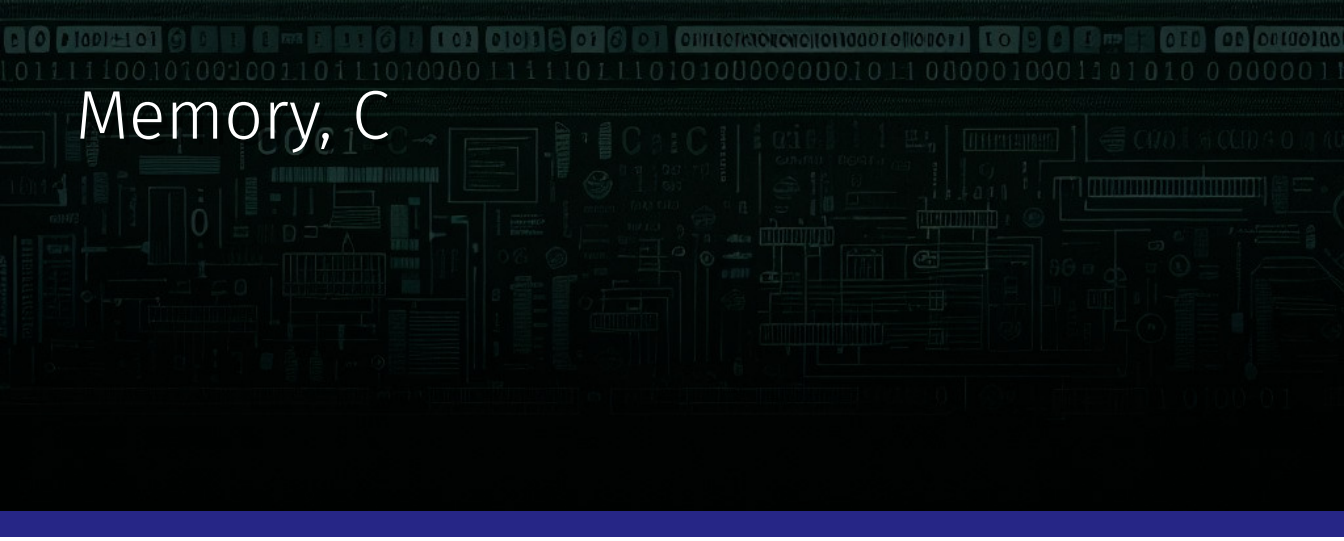

Memory, C

CS 2130: Computer Systems and Organization 1
November 17, 2025

Announcements

- Homework 8 due tonight on Gradescope
- Homework 9 available this afternoon
- Lab 10 tomorrow: Memory Errors
- Lab 11 next Tuesday (can check off for full credit by 12/5)

The Heap

The heap: unorganized memory for our data

- Most code we write will use the heap
- *Not a heap data structure...*

The Heap: Requesting Memory

```
void *malloc(size_t size);
```

- Ask for `size` bytes of memory
- Returns a `(void *)` pointer to the first byte
- It does not know what we will use the space for!
- Does not erase (or zero) the memory it returns

The Heap: Freeing Memory

Freeing memory: `free`

```
void free(void *ptr);
```

- Accepts a pointer returned by `malloc`
- Marks that memory as no longer in use, available to use later
- You should `free()` memory to avoid *memory leaks*

Garbage

Garbage - memory on the heap our code will never use again

- Weird: defined in terms of the future!
- Compiler can't figure out when to free for you

Garbage

Garbage - memory on the heap our code will never use again

- Weird: defined in terms of the future!
- Compiler can't figure out when to free for you

What about Java?

Garbage Collector

Garbage Collector - frees garbage “automatically”

- **Unreachable memory** - memory on heap that is unreachable through pointers on the stack (or reachable by them)
  - Subset of all the garbage
  - Identifiable!
- Takes resources to work
- *Very* popular - most languages have garbage collectors
  - Java, Python, C#, ...

Common Memory Bugs (reading)

List example

More on `man` pages

```
char *strsep(char **s, const char *d);
```