C, Memory, string.h

CS 2130: Computer Systems and Organization 1

April 17, 2023

Announcements

- Homework 8 due tonight at 11pm
  - Limited number of submissions, test your code before submitting
- Exam regrade requests due by Friday

```
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

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 - 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 - 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#, …

`man` pages, finding `man` pages

`malloc` man page

List example

Common Memory Bugs (reading)

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