CSCI 4061 Discussion 7

3/5/18

Overview

- Review of malloc and free
- Stack smashing
- Exercise

malloc and free

- Memory allocated using malloc is placed on **the heap**, which **does not get reclaimed at the end of a function**.
- This memory needs to be freed using free(void*).

Stack Smashing

- A memory error resulting from writing memory outside the current function's stack space.
- This potentially corrupts the stack of another function call.

Stack Smashing Example

```
// We don't check the bounds of bar.
void foo(char* bar) {

    char barbar[12];
    strcpy(barbar, bar);
}
```

Viewing Memory Usage in Top

- Simply type ‘top’ into a terminal to view processor and memory usage for currently running applications.

Exercise

- In the files for today, some code is given which has several memory issues.
- **You may only make changes involving memory allocation and deallocation.**

- Fix all of the issues so that:
  - No segmentation faults occur
  - No stack smashing occurs
  - There are no memory leaks (view in top)