ITBU 373, Operating Systems Homework 10

Chapter 14, *Operating Systems, Three Easy Pieces*

Readings

Read chapter 14 in the *Operating Systems, Three Easy Pieces* book.

Discussion Questions

Answer the discussion questions in writing.

1. What does it mean to say that the stack is allocated *implicitly*? Why is the stack called *automatic memory*?
2. What does it mean to say that the heap is allocated *explicitly*?
3. When you call `malloc()`, the call results in memory allocations to both the stack and the heap. Why does this happen? What does the memory allocated from the heap contain? What does the memory allocated from the stack contain?
4. What are the parameters to `malloc()`? What are the return values to `malloc()`?
5. (not in book) What are the parameters to `calloc()`? What are the return values to `calloc()`?
6. Why is it a bad idea to use a numeric literal as an argument to `malloc()`?
7. What is the purpose of casting the return value of `malloc()` to a type of pointer? Why does the book say that such a casting "doesn't really accomplish anything?"
8. What is the purpose of the call to `free()`?
9. What does the *garbge collector* do? Notice that the languages build on top of C (such as Java, C++, and C#) all have garbage collectors. Do you think it's better to have a garbage collector or not? Justify your opinion (briefly, don't make a long philosophical speech!).
10. Briefly describe the following errors:
    (a) forgetting to allocate memory
    (b) not allocating enough memory
    (c) not initializing memory
    (d) forgetting to free memory
    (e) prematurely freeing memory
    (f) freeing memory multiple times
11. What is a *dangling pointer*?
12. What is a *memory leak*?
13. Are `malloc()` and `free()` system calls? Why or why not?