

Class 33: Pointers and Arrays

Held: Friday, 29 October 2010

Summary: We continue our exploration of pointers in C. In particular, we consider the relationship between pointers and arrays in C.

Related Pages:

- EBoard.
- Lab: Pointers, Revisited.
- Reading: K&R 5.3-5.4.

Notes:

- We may have propsies today. Yay!
- EC for MD's Monday presentation.
- EC for Saturday's Men's Soccer game.
- EC for LoC truck.
- Reading for Monday: K&R 5.3 and Appendix B.
- Are there questions on Exam 2.

Overview:

- Pointers and Arrays.
- Labs.

Pointers and Arrays

- In C, a pointer and an array are more or less the same (at least internally).
- In fact, you can (mostly) treat a pointer as an array or an array as a pointer.
- Some key differences:
 - When you declare arrays, you usually specify the size.
 - You should not be able to change where an array "points".
- In many ways, arrays are "syntactic sugar" for pointers.
- For example `a[i]` is really just `*(a + i)`.

Your Questions

- Can a pointer point to a pointer? If so, can you access the value by using multiple `*` commands?
- In page 105 about the comparison between the pointers and arrays, what is the difference in the declaration of arrays and pointers?
- Please explain `(*t = *s) != '\0'`. Thanks very much.

- What is a cast?
- I don't understand what the point of allocp and afree are. Can you just talk about them a little?
- Could you provide some examples in class illustrating the situations where it would be prudent to use pointers in a program?

Labs

- Continue the lab on pointers.
 - Begin the lab on pointers, revisited.
-