

Class 31: Pointers

Held: Tuesday, 26 October 2010

Summary: We study pointers, one of the important data types in C.

Related Pages:

- EBoard.
- Lab: Pointers.
- Reading: K&R 5.1-5.2.

Notes:

- Cool Theatre stuff on campus this week.
- Reading for tomorrow: K&R 5.3-5.4.
- Don't forget questions!
- LoC Truck on Friday.
- We'll go over some questions and issues on HW6.

Overview:

- Your Questions.
- Lab.

Memory and Pointers

- “Everything” (or at least every variable) has to be stored somewhere in memory.
- The program needs to know where it's stored.
- In C, you can access information about where things are stored.
 - In Scheme (at least standard Scheme), you cannot.
- The address-of operation (&) tells you where in memory that thing is stored.
- An address is also called a *pointer*.
- The type of a pointer is specified by the type of the thing pointed to, followed by a *. So `int *` means “a pointer to an integer”.
- You can find the contents of something pointed to by prefixing the pointer with *.

Your Questions

- Couldn't you also return the new a and b instead of making swap a void function?
- In the swap routine example, are pointers the only way to make that function work?
- I thought the main advantage in using pointers is to decrease the use of memory, but I don't see that in the examples. Why do we need pointers?

- Could you explain the * operator? It seems clearly laid out on page 94, but I'm just not wrapping my head all the way around it.

Lab

- Do The Lab.
 - Be prepared to reflect.
-