Pointers/Multidimensional Arrays

10-17-2006

Opening Discussion

- Midterm results.
- Do you have any questions about the reading?
- What did we talk about last class? Let's go look at the code because that was a while ago.
- Do you have any questions about the assignment?

Pointers

- We have talked about basic types in C and last time we introduced the idea of an array type. We can also create variables of pointer types.
- The name really says what a pointer is. A pointer variable points to some other part of memory. Technically it stores a memory address and you can follow that address to get to the memory that it refers to.
- Following a pointer to see what it points at is called dereferencing.
- You can make a pointer type of any type and that is a new type. As such, you can make pointers to pointers and so on.

Pointer Syntax

- We declare a pointer variable using *. You saw this when we declared our FILE * variables for files. The variables fin and fout were pointers to a FILE. Put a * between the type and the name and you get a pointer. You can put more than one * in there as well.
- If you want to get the address of a memory location use the & operator. The & operator adds a * to the type.
- If you want to follow a pointer you use the * syntax. So the * operator removes a * from the type. (How is that for confusing?)
- a[i]==*(a+i)

Pass by Reference

- We have had times when we wanted to return multiple values from a function, but the C syntax only allows a single return type.
- To get around this we pass in pointers. This is commonly called “pass-by-reference”. What we have normally done is called “pass-by-value”.
- When you pass something by reference that pointer is passed by value, but you can follow the pointer and modify the original memory. (Swap function)
- 1-D arrays are basically pointers with memory allocated. All arrays are passed as pointers.
- Helpful but problematic.

Multidimensional Arrays

- Use as we can make pointers to pointers, we can also make arrays of arrays. These are multidimensional arrays.
- What does the syntax of a multidimensional array look like?
- Multidimensional arrays are not pointers to pointers (despite very similar syntax). What must you do that is special when you pass a multidimensional array?

Searching Arrays

- If I give you an array and ask you to find a particular value, what will you do? How many elements will you have to look at on average?
- If I tell you the array is in sorted order what can you change in your search? How much more efficient would this method be?

Minute Essay

- We covered a lot of stuff today, including a lot of syntax. Do you have any questions about this material?
- Remember that assignment #4 is due on Thursday.