

## Class 36: Multi-dimensional arrays

**Held:** Wednesday, 3 November 2010

**Summary:** We consider multi-dimensional arrays. That is, we consider arrays of arrays. We examine how one declares them, why one might use them, and some complexities in their use.

### Related Pages:

- EBoard.
- Lab: Multi-Dimensional Arrays.
- Reading: K&R 5.6-5.9.
- Due: Exam 2.

### Notes:

- I'm trying to reserve most of today for lab.
- There is no reading for Friday.
- I hope you enjoyed yesterday's digressions. I do think that it is important that prospective computer scientists know about these "big ideas" from the discipline.
- As part of PragProWriMo, I'm trying to write extended pieces on the more complicated parts of K&R and C. If you observe things that you'd like a short piece (2-4 pages) on, please let me know.
- We'll use a few minutes at the start of class for the final set of questions on Exam 2.

### Overview:

- Multi-dimensional arrays.
- Why?
- Arrays of pointers vs. Arrays of arrays.
- K&R example, revisited. (Reading forthcoming.)
- Lab.

## Multi-Dimensional Arrays

- You can make arrays of anything.
- Even arrays of pointers and arrays of other arrays.
- What's the syntax for declaring?
- An array of pointers

```
TYPE *NAME[SIZE];
```

- An array of arrays

```
TYPE NAME[ROWS][COLS];
```

- You can initialize them when declaring.

```
char *strings[3] = { "foo", "bar", "baz" };
int stuff[2][3] = { { 1, 2, 3},
                   { 4, 5, 6} };
```

## Why Do We Need Multi-Dimensional Arrays?

- When we have arrays of strings.
- When we want to represent matrices for computation (e.g., for those physics types)
- When we want to represent compound data, one way to represent compound data is an array. (E.g., we might represent a student in this class by a five element array of strings: Last name, First name, Student ID, Major, and Grade.)
  - A collection of compound data can then be represented as an array of arrays. (Or, more precisely, an array of arrays of arrays.)
- As you've no doubt determined, tables of numbers are often useful. Two dimensional arrays simulate tables.

## Arrays vs Pointers, Revisited

- We've seen that arrays and pointers are similar in some ways, but different in others.
- What's the difference between an array of pointers and an array of arrays?

## Selected Issues in the Example

*This section of the class is being moved to a forthcoming reading that I will distribute after we discuss sorting.*

- K&R talk about sorting, but raise other interesting issues.
- When you're reading strings, you need to read them into somewhere.
- They previously wrote `alloc` to give you a way to deal with that issue.

## Lab

- Start The lab.
-