Tip 18: Declare arrays when you know their size

Ben Klemens

6 November 2011

level: still basic
purpose: save the memory register stuff for when you really need it

You can allocate arrays to have a length determined at run time.

I point this out because it's easy to find texts that indicate that you either know the size of the array at compile time or you've gotta use `malloc`. But it's perfectly fine to delay initialization of the array until you find out its size. [Again, this is the difference between C in the 1970s when this either-or choice was real, and the C of this millennium.]

For example, here's a program (found via One Thing Well[1]) that will allow you to run several programs from the command line in parallel[2]. The intent of the following snippet (heavily edited by me) is to get the size of the array from the user using `atoi(argv[1])` (i.e., convert the first command-line argument to an integer), and then having established that number at run-time, allocate an array of the right length.

```
pthread_t *threads;
int thread_count;
thread_count = atoi(argv[1]);
threads = malloc(thread_count * sizeof(pthread_t));
```

This is fine, and I don't mean to disparage the author when I rewrite this, but we can write the edited lines of code above with less fuss:

```
int thread_count = atoi(argv[1]);
pthread_t threads[thread_count];
```

There are fewer places for anything to go wrong, and it reads like declaring an array, not initializing memory registers.

The original program didn't bother freeing the array, because the program just exits. But if we were in a situation where the first would need a `free` at the end, the variable-length initialization still doesn't need it; just drop it on the floor and it'll get cleaned up when the program leaves the given scope.

By the way, you can find people online who will point out that manually allocated memory is faster than automatic. I recommend not caring. The speed difference is a

[1] `http://onethingwell.org/post/9960491695/parallelize`
[2] `http://www.marco.org/2008/05/31/parallelize-shell-utility-to-execute-command-batches`

few percent, not an order of magnitude (and is architecture- and compiler-dependent whether there's any difference at all). It's a subjective thing, but in this case I'll gladly trade more readable code for the small speed gain, if any.