

Tip 86: End tip mode

Ben Klemens

22 March 2012

Hey, thanks for reading. It was a fun challenge to produce a new tip every other day. I don't envy columnists whose job is to say something interesting every Monday, Wednesday, and Friday.

I'd resolved to keep at it until I reached a certain goal, which I reached just a few days ago (hint: see tip #83 on m4 macros to generate docbook XML (Entry #133)). I'll let you know when there's a publication date.

So what's next for this blog? My real interest is in better modeling. I've blogged this before (Entry #014), and it still holds: I'm one of the people who learned a ton of computing technique for the purpose of going somewhere with it, and for me that somewhere is models of the world that go beyond simple linear fare for something more elaborate.

Multilevel models (which go under *several* other names) are built around taking one model and using its output as an input to another model. That seems like a simple enough sentence, but you're not going to get there until you can generate objects with the appropriate functions in them (Entry #104), and you may further need to conveniently extend an existing struct (Entry #113). Now that you're doing one potentially computationally expensive operation nested inside another, both had darn well better be fast, so you had better know how to thread (Entry #115) and how to speed up your database (Entry #127). You're not going to write all this from scratch, so you'll need to know how to use libraries (Entry #051), which you can't do unless you can write a decent makefile (Entry #050).

It may not be evident, but I'm human, and I want to have an easy time at the computer, so within C's grammar, having functions that can take in lists of arbitrary length (Entry #076) and functions that take named and optional arguments (Entry #100) saved my life. Partially-featured debuggers are like walking around a dark room with sunglasses on (`R debug ()`, I'm looking at you), so you'll be using GDB and you'll need to know how to print all these elaborate structures in the debugger (Entry #065). Having a decent shell (Entry #086) and knowing how to use it will also make all this less painful.

And you'd darn well better well have a decent documentation system (Entry #117), or else you'll get lost in your own maze.

So that's how I wound up here. The goal has always been the same: I want to be able to go beyond computationally cheap models and more accurately portray the world around us. It's 2012, and you'd think that we'd be able to do that without knowing anything about all this low-level stuff, but we're not there yet. Maybe the stuff I'm

working on will help us to get there. I'll write more on the goal of better modeling in future, sporadic blog posts.