

# Random idea of the week: Parallel compiler

Thomas Paul Carlson (tpc4@kent.ac.uk)

22nd March 2007

## 1 Stuff.

I've had some fun messing around making `htmlangs` first ever compiler, and although it is incredibly messy and hacky it does actually produce some working code now. When writing it, it struck me that what I was dealing with a lot of the time on the parser end was streams of characters. I (Along with a bunch of other 2nd/3rd year compsci students) have been doing the CO631 Concurrency module which introduces us to the idea of processes and networks and programming in parallel. So why not apply the same thing to a compiler?

I googled around a little bit and found that the early occam compilers were in fact written in occam. Which is pretty neat, if you ask me. Some more googling brought me to a mailing list archive discussing bits and pieces...apparently these early occam compilers written in occam had various processes to do bits of the compilation. For example, there was a process network to handle type-checking, another network to do tokenizing of the input stream and another to do code generation. The way I see it, with a simple imperative language (Think java minus most of its features) you could spawn a new process for every line of code to do type checking, once you knew which symbols matched up with what type.

I don't really know how I would represent a tree in occam at all. Since most of what I know about compilers is basically tree transformation, this is kind of important. Hum. Something to come back to later, I think...

Getting stuff from the outside world to occam-world is pretty easy. Pipe some catt'ed stuff to the program and get it to read stuff coming in on the "keyboard" (Really standard input) channel. If working with a java like program it'd be reasonably easy to break the program down into blocks by breaking on spaces (With special stuff for strings and things like that).

Anyway, these are just random ideas. Maybe I'll have a go at making a compiler-thing in occam some day. Sounds like a very shiny thing that should be at least attempted!