c++ in linux kernel

By: **Linus Torvalds** (torvalds.delete@this.linux-foundation.org), June 5, 2010 7:07 pm Room:

Heath Provost (galvanash@hotmail.com) on 6/5/10 wrote:
>
>As for C++ exceptions - the same thing really applies here.
>They are trying to write explicit code. Exceptions are the
>poster child for implicit magic...

Yes, exceptions is a good example. The Linux kernel actually does its own exception mechanism, exactly because that way we control what is going on (and do it much more targeted to the actual need in question while giving much better performance and avoiding the crazy unwinding issues).

And I really do dislike C++. It's a really bad language, in my opinion. It tries to solve all the wrong problems, and does not tackle the right ones. The things C++ "solves" are trivial things, almost purely syntactic extensions to C rather than fixing some true *deep* problem.

(The C++ objects, templates and function overloading are all just syntactic sugar. And generally *bad* syntax at that. And C++ actually makes the C type system actively *worse*.)

In non-systems programming, you should almost certainly use a language that offers garbage collection. That will possibly make a real difference in the complexity of your application. The C++ features? Largely useless, and just helps you screw up more.

And in systems programming, you're simply better off with C. You'll have a way easier time using all the existing code and libraries out there (re-using C++ code? Good luck). Fewer headaches, fewer opportunities to mess up the design and pick some unstable template library.

So in neither case is C++ likely the right choice.

Linus