Subject: Re: C++ FQA
Posted by cbpporter on Mon, 12 Nov 2007 22:22:28 GMT
View Forum Message <> Reply to Message

Quote:
Sorry for being rude, but your understanding of actual GC (and especially conservative GC) is sort of lacking.

That could very well be the case, but I believe I have some if not enough teoretical and practical experience. What exactly did I say that was inaccurate? I was referring to generic garbage collection (no ref-counting though), but biased toward mark and sweep style algorithms, not exactly to D's implementation (about which I have only superficial knowledge). And what do you mean by conservative GC?

Quote:
I am no expect in GC and perhaps I am wrong about this, but IMO generational GC and moving GC are mutually exclusive.

Well a moving GC (which makes little sense without a compacting GC) moves memory chunks around when neaded and modifies pointers to point to the new locations. A generational GC simply optimizes the allocated objects list so that objects that were created recently are faster to deallocate than old objects (plainly put), because newly created objects have a larger change to get destroyed. So I don't see any reason for them to be mutually exclusive. I could be wrong though.

Quote:
And no, you cannot have destructors and GC working together.

Well you can. With a little extra care, you can have fully functional destructors (just be sure never to physically deallocate memory, just do cleanups). But with GC you rarelly need non-trivial destructors. And if the programing language has a "scope" clause like D, things get a lot simpler.

Quote:
I believe that with a couple of tricks, I am getting (with U++) more than I could get with GC - all resources are managed by program structure.

No arguing here. This is one of the main reasons I like U++. But I would say "almost all", because once in a while you do need to manually manage memory outside of program structure. But those cases are insignificant to overall code size and scope.