

Eliminating implicit function declarations

Thursday, September 23, 2021 7:00 AM (30 minutes)

The 1999 revision of ISO C removed implicit function declarations from the language. Instead, all functions must be declared (with or without a prototype) before they can be called. In previous language versions, a function `f` was implicitly declared as `extern int f ();` if the identifier `f` was used in a call expression (such as `f(1, 2, 3.0)`).

When GCC switched the default to C99 mode, it was impossible to disable implicit function declarations by default because too many `autoconf` checks (and similar compile-time inspection) failed, which often resulted in successful compilation (and testing) of programs without the intended feature set.

Over the years, not much progress has been made on this issue. For example, a GCC configure test was only fixed in 2019. Recently, Apple Xcode enabled `-Werror=implicit-function-declaration` by default, but apparently without fixing resulting problems across free software upstreams.

This session intends to explore whether it is time to make a concerted effort at this problem, and how to approach it.

Intended format: short **prepared presentation** (10 minutes), followed by discussion.

I agree to abide by the anti-harassment policy

I agree

Primary author: WEIMER, Florian (Red Hat)

Presenter: WEIMER, Florian (Red Hat)

Session Classification: GNU Tools Track

Track Classification: GNU Tools Track