

Summary

Recursive Make Considered Harmful by Peter Miller discusses software construction using Make for projects that span several directories. Using an example, Miller shows the shortcomings and dangers of the traditional approach where each directory has its own Makefile that is executed by an recursively invoked instance of Make. In the example, two directories have mutual dependencies which cannot be separated in two Makefiles without effectively writing one global Makefile. Distributed Makefiles are notorious for missing these inter-directory dependencies and thus causing incorrect builds. The traditional counter measures fail to solve the problem reliably or lead to overlong build times because the build process is iterated for several times. As a consequence, Miller advocates the use of a single, global Makefile that knows all dependencies. This solution is traditionally greeted with skepticism and Miller rejects often-used arguments against them: a single Makefile is not necessarily bigger than multiple Makefiles, less maintainable, more complicated to create, or causes longer build times.

Miller devotes a large part of the paper to efficient Makefiles. He concluded from experiments that time spent by Make itself is significant, especially caused by frequent re-evaluation of variables. Providing detailed code he discusses how to store automatically derived dependencies for C programs in many files such to avoid recomputing dependencies unnecessarily. The paper ends with the finding that previous literature on Make fails to discuss the usage of Make in multi-directory projects.

The paper offers valuable advice for practitioners: the running example, the shortcoming of the traditional approach, and the advocated solution are discussed in detail. From a scientific point of view the paper is less satisfying. It contains many claims, especially about the performance of Make, that are not substantiated. An exact and reproducible breakdown of how much time is spent where in a Make-based build process would have been useful. The paper is written under the premise that Make's design is not to blame for the problems observed in large builds. When the paper was written in 1998, already a number of attempts existed (tools like Mk, Cake, NMake) to overcome the weaknesses of Make. From a scientific point of view, a comparison with these would have been interesting.