Implicit Undefined Behavior in C

David Keaton

2018-04-25

After recent discussions, it has become clear that it would be useful to reacquaint ourselves with the concept of implicit undefined behavior in the C standard.

Definition of Undefined Behavior

Subclause 3.4.3, paragraph 1, states the definition of undefined behavior (emphasis added):

> behavior, upon use of a nonportable or erroneous program construct or of erroneous data, *for which this document imposes no requirements*

By this definition, if the document imposes no requirements, the behavior is undefined. If the document does not mention a subject, it is not imposing any requirements, so the behavior is undefined. Although this is sufficient, the concept is further amplified in the conformance clause.

Conformance Clause

Clause 4, paragraph 2, explains how undefined behavior is indicated (emphasis added):

> If a "shall" or "shall not" requirement that appears outside of a constraint or runtime-constraint is violated, the behavior is undefined. Undefined behavior is otherwise indicated in this document by the words "undefined behavior" *or by the omission of any explicit definition of behavior. There is no difference in emphasis among these three; they all describe "behavior that is undefined".*

With this explanation, it is clear that the definition means to include unmentioned behavior in the class of behavior called undefined. This understanding is further cemented in the response to a defect report.

DR 017

In 1990, when DR 017 was submitted as an X3J11 interpretation request, the above two quotes appeared together in the definition of undefined behavior. The second part was later moved to the conformance clause as shown above. The DR response states:

> There are many places in the C Standard that leave behavior implicitly undefined. The Committee chose as a style for the C Standard not to enumerate these places as explicitly undefined behavior. Rather, subclause 3.16, page 3, lines 12-16 explicitly allow for implicitly undefined behavior and explicitly give implicitly undefined behavior equal status with other forms of undefined behavior.

This explanation makes it clear that the text of the standard matches the intent.