

从“简单命令解析状态机”入手：浅析 goto 语句面对复杂控制流时的可维护性代价

顾明宇

2025.10.28

摘要

goto 语句曾因其强大的跳转能力而备受青睐。然而随着 E. W. Dijkstra 的著名文章《Go To Statement Considered Harmful》[1] 的发表，它成为了结构化编程运动的主要批判对象，并在现代编程实践中被严格限制。本文旨从“简单命令解析状态机”为案例引入，具象化分析 goto 语句在复杂控制流中引发的可维护性灾难问题。

本文通过“简单命令解析状态机”案例，采用对比实验方法，针对“start->run->stop”命令解析需求，分别用结构化编程（switch-case）和 goto 语句两种方式实现 C 语言代码，并通过三次需求变更（增加 pause 状态、错误重试机制、位置相关错误信息）来对比分析两种实现方式的维护性差异。研究表明，goto 版本在每次需求变更时都需要修改多个跳转标签，代码复杂度呈指数级增长，而结构化版本仅需在相应状态分支中添加逻辑，修改成本低且易于维护。基于圈复杂度和可维护性指数的量化分析显示，goto 版本的圈复杂度从初始的 4 增长到最终的 10，而结构化版本仅从 4 增长到 6。本研究以此为现代 C 语言编程实践提供参考，证明了遵循结构化编程原则对提高代码可维护性的重要意义。

参考文献

- [1] EW Dijkstra. Letters to the editor: go to statement considered harmful. *Communications of the ACM*, 11(3):147–150, March,1968.