

Switch statement fdo reordering

Edmar Wienskoski

Your affiliation

`your-address@example.com`

Abstract

When gcc parses a switch statement, it uses some criteria to decide between generating a jump table or a binary search tree. The jump table have a fixed significant cost, and for large switch statements the sequence of compare-branches from the root to the leaves can also be costly.

The optimization described here, collects a histogram of a switch statement condition expression and uses it to balance the binary search tree. We also implement a default case promotion: Single values in the histogram that maps to default in the switch statement, are candidates to become a new node in the binary search tree, which give a chance to that particular value to have its path on the tree optimized.