

Copy-and-Patch Compilation

Ondřej Krejčí

xkrejtc69@stud.fit.vutbr.cz

Oliver Kuník

xkunik00@stud.fit.vutbr.cz

Abstract

Copy-and-patch is an experimental approach to compilation which produces output binary from precompiled snippets of code. Its goal is to produce code faster than other compilation techniques and produce more effective code than non-optimizing compilers.

It generates code using precompiled snippets of binary code implementing simple operations called stencils. These stencils contain holes which are filled with values like addresses or constants during compile time. The stencils are available to the compiler as a library. During compilation the compiler selects a stencil to implement an operation from the source program. It then copies this stencil to the output (*copy*) and fills in the holes with appropriate values from the source program (*patch*).

The stencil library itself is generated beforehand by a compiler called MetaVar. This compiler generates stencils from templates describing a single operation. Single template can generate multiple stencils with different configuration of parameters for the operation (e.g. different registers or locations on stack).

The article introducing copy-and-patch compilation gives two examples demonstrating its use, a WebAssembly compiler and a high-level metaprogramming language. The WebAssembly compiler is compared to both non-optimizing and optimizing compilers. The high-level language is used to create an SQL query compiler.