

Objectify assembler output in the Maxine client compiler

Currently the Maxine client compiler directly emits assembly instructions from its LIR (low-level intermediate representation) intermediate language. This is unsuitable for further processing, i.e. applying code improvement transformations on it, and requires a per-platform LIR implementation. The task is to extend previous work in this area: already an OO layer between LIR and code output has been implemented, working for simple test programs. Real benchmarks do not work yet though. Further there is need for a short analysis regarding performance and memory consumption.

Goals

- implement some transformations enabled by this intermediate representations

Tasks

- get to know Graal source code
- inspect and find the remaining bug(s) in the current implementation
- evaluate performance and memory consumption
- implement some transformations enabled by this intermediate representation

Possible extension to Bsc, Msc thesis

- instruction scheduling using this representation (see other proposal)
- multiplatform support

Other information

- programming language Java

Further information

- Thomas Schatzl, HF305, thomas.schatzl@jku.at