JOHANNES KEPLER
UNIVERSITY LINZ

Dipl.-Ing. David Leopoldseder
Institute for System Software

T +43 732 2468 4356
F +43 732 2468 4345
david.leopoldseder@jku.at

Secretary:
Birgit Kranzl
Ext 4341
birgit.kranzl@jku.at

Master Thesis

Truffle CIL Interpreter

Student: Patrick Hagmüller
SKZ/Matr.Nr.: 921 / 0956547
Email: patrick.hagmueller@gmx.at

Advisor: Dipl.-Ing. David Leopoldseder

Start date: 01.04.2018

Truffle [1,2] is a programming language implementation framework on-top of the GraalVM. Language implementers write an abstract syntax tree (AST) interpreter, which is registered in the Truffle eco-system to be part of the polyglot GraalVM world.

There are already several language implementations in the Truffle eco-system [2], from imperative languages like C to functional ones like Python.

The goal of this thesis is to write a Common Intermediate Language (CIL) interpreter for Truffle, focusing on creating a Truffle interpreter for textual CIL code.

The thesis should also determine the set of nodes that are needed for modeling the semantics of the CIL code.

The goals of this thesis are:

- Implementation of a Truffle CIL Interpreter including a parser and Truffle nodes that can be partially evaluated like other bytecode-based languages on-top of Truffle[4].
- Implementation of a subset of CIL such that the interpreter can be tested on established C# benchmarks.
- Implementation of an Object Model based on the Truffle Object Model.
- Evaluation of the implementation with a set of non-trivial benchmarks.

Modalities

The progress of the project should be discussed at least every two weeks with the advisor. A time schedule and a milestone plan must be set up within the first 3 weeks. It should be continuously refined and monitored to make sure that the thesis will be completed in time. The final version of the thesis must be submitted not later than xx.xx.xxxx.

[1] http://openjdk.java.net/projects/graal
[2] Würthinger et all. “One VM to rule them all”
[3] http://www.ssw.uni-linz.ac.at/Teaching/Lectures/Sem/2003/reports/Fischereder/Fischereder.pdf
[4] Rigger et all. “Sulong, and Thanks for all the Bugs: Finding Erros in C Programs by Abstracting from the Native Execution Model”
[5] Wöß et all. “An object storage model for the truffle language implementation framework”