An extension of embedded C for parallel programming

Master-Thesis von Bastian Gorholt
August 2014

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Informatik
Software Engineering Group

An extension of embedded C for parallel programming

Vorgelegte Master-Thesis von Bastian Gorholt

1. Gutachten: Sebastian Erdweg
2. Gutachten:

Tag der Einreichung:

1 Introduction

- mbeddr is a language and ide for embedded programming that facilitates programming by providing higher level language mechanisms
- parallel programming is becoming ever more important
- yet, mbeddr is still lacking higher-level support for parallel programming
- compared to library extensions language based support for pp features multiple benefits: individual problem-specific syntax, reduction of erroneous input by type system enhancements, program-wide optimization of generated code
- short examples for benefits
- ParallelMbeddr introduces task-based explicit parallel programming with shared memory that can be synchronized
- Objectives: apropriate syntax, type system support for safer code, optimized output
- outline of the work

Kommentar von Bastian
Question: Introduce example that is later used for the evaluation already here for motivational reasons?

2 Basics

- what is parallel programming?
- shared memory vs. message passing
- threads, mutexes, synchronization
- explicit vs. implicit parallel programming
- embedded programming objectives: limited memory, time boundaries like WCET (not part of this work)

3 Analysis

3.1 Existing work

Introduce parallel programming concepts of other languages and how they relate to this work.

3.2 Language Design

New syntax, typing rules and generation output for:

- tasks
- futures
- sychronization primitives

Kommentar von Bastian
Question: Where to put the reasoning behind each language construct?

4 Implementation

4.1 Mbeddr

short introduction into the workflow with concepts and aspects

4.2 Tasks and Futures

for all items mentioned in the language design and necessary intermediary MPS concepts:

- implemented structure
- implemented typing rules
- implemented generation rules

Kommentar von Bastian
Question: How precise shall I be?

4.3 Synchronization

see above

4.4 Optimization

- short introduction into general cbmc workflow
- every accomplished optimization in a subsection

5 Evaluation

- show if and how objectives mentioned in the introduction are met by the implementation
- use ilustrating examples

Literaturverzeichnis