
Bachelor/Master Thesis Topic

Diagnosis and Localization of Memory Leaks

Motivation and Background

Memory leaks are latent defects triggering memory depletion during runtime. They are usually caused by forgetting to free allocated heap space (C/C++) or to remove all references to obsolete objects (Java). Even if software is known to contain memory leaks identifying their root causes (so-called defect localization) can be tricky. Causes for this are manifold: leaks manifest slowly (i.e. they need prolonged execution time to surface); data structures like caches can keep allocated memory for legitimate reasons; it requires a variety of data to decide whether an object/memory fragment is obsolete or needed. Several approaches and tools for localization of memory leaks exist [1, 2, 3]. While some suffer significant performance issues [1], other require sophisticated static analysis [2] or are proprietary (e.g. LeakBot from IBM). A simpler approach [3] using dynamic analysis exploits regression testing.

Goals

This project will compare and evaluate several approaches for localization of memory leaks based on runtime analysis, and attempt to improve their performance and possibly also accuracy.

Description of the Task

The specific tasks are:

- Understand selected approaches for localization of memory leaks and try to improve them
- Create a tool for memory leak detection (C or Java) based on investigated approaches or a subset of them
- Perform experimental evaluation and comparison of accuracy of selected approaches or their modified versions on an exemplary application with memory leaks.

Research Type

Theoretical Aspects: *****

Industrial Relevance: *****

Implementation *****

Prerequisite

The student should be enrolled in the bachelor/master of software engineering/informatics program, and has completed the required course modules to start a bachelor/master thesis.

Skills required

Programming skills in Java or C++, understanding of, or willingness to learn, the software engineering methods and of memory management mechanisms and monitoring tools needed for the project.

References

[1] J. Clause and A. Orso. LeakPoint: Pinpointing the Causes of Memory Leaks. ICSE 2010.

[2] M. Hauswirth and T. M. Chilimbi. Low-overhead memory leak detection using adaptive statistical profiling, ASPLOS 2004.

[3] Felix Langner and Artur Andrzejak. Detection and Root Cause Analysis of Memory-Related Software Aging Defects by Automated Tests, MASCOTS 2013.

Contacts

Lars Grunske, Humboldt-Universität zu Berlin, Institut für Informatik, Lehrstuhl Software Engineering, Unter den Linden 6, 10099 Berlin, Germany

Application

Please contact me during my office hours or send me an email with the title: "[ThesisProject]-DLML" to se-career@informatik.hu-berlin.de