



# Bachelor's/Master's Thesis

## Testing the Impact of Hostile Environments on Software Stability

### Background

In order to run efficiently on a large variety of different platforms and to provide features tailored for various use cases, most software systems are highly configurable. A common way for configuring an application to perform well in a specific execution environment is reading *environment variables*. For instance, the content of the CPATH environment variable affects the search path of the gcc pre-processor when it is invoked.

Unfortunately, the reliance on environment variables for software configuration and context-aware adaptation also poses several challenges to software quality assurance. While the problem of *debugging* misconfigurations once they occur has been addressed in a number of research articles, the question how to detect such misconfigurations in the first place has not been adequately addressed.

### Objectives

The goal of this thesis is to develop two approaches for configuration testing, both of which can be based on *elektrify-getenv*, an existing automated approach to intercept getenv invocations (POSIX's method to read environment variables) in binaries. The first approach should follow a random testing scheme to serve as a baseline to compare the second approach against. The second approach can follow one of several possible strategies. It can be either based on a model of (il)legitimate values derived from observed or specified configurations, implement a learning based approach that selects new tests based on the results of earlier tests, or rely on code analyses to calculate the likelihood of provoking a software failure.

### Prerequisites

Candidates should have basic familiarity with Linux/UNIX programming (preferably in C) and fundamental knowledge of software testing. The thesis will be written in English.

### Start

Immediate

### Contact

**Dr. Stefan Winter**  
sw@cs.tu-darmstadt.de  
+49 6151 16 25226  
S2|02 E221

#### Literature



#### Analysis



#### Implementation



#### Awesomeness

