

Towards the Property-Based Testing of an L4 Microkernel API

Cosmin Dragomir

Faculty of Automatic Control and Computers
University POLITEHNICA of Bucharest
Splaiul Independentei nr. 313
Sector 6, Bucuresti, Romania
cosmin.dragomir@cti.pub.ro

Mihai Carabas

Faculty of Automatic Control and Computers
University POLITEHNICA of Bucharest
Splaiul Independentei nr. 313
Sector 6, Bucuresti, Romania
mihai.carabas@cs.pub.ro

Lucian Mogosanu

Faculty of Automatic Control and Computers
University POLITEHNICA of Bucharest
Splaiul Independentei nr. 313
Sector 6, Bucuresti, Romania
lucian.mogosanu@cs.pub.ro

Razvan Deaconescu

Faculty of Automatic Control and Computers
University POLITEHNICA of Bucharest
Splaiul Independentei nr. 313
Sector 6, Bucuresti, Romania
razvan.deaconescu@cs.pub.ro

Nicolae Tapus

Faculty of Automatic Control and Computers
University POLITEHNICA of Bucharest
Splaiul Independentei nr. 313
Sector 6, Bucuresti, Romania
nicolae.tapus@cs.pub.ro

Software testing has been a significant part of the software development process for the last 30 years and is gaining even more importance with the increasing complexity of software products. As each application has its own requirements, multiple software testing methodologies exist. It is the decision of the developers to choose the best suited types of testing methodologies for their product. This paper presents the design and implementation of a property-based testing framework. Unlike traditional testing methods this methodology uses the formal specification of the API to automatically generate the input and validate the output. The framework will be used to test the API of an L4 microkernel (called VMXL4); VMXL4 possesses the constraints of an embedded environment and of an ongoing development of a stateful system.

Property-Based Testing, L4 Microkernel, API

1. INTRODUCTION

The software industry has been constantly growing in the last decades and the liability and robustness of the software products must match their requirements in order to remain competitive. To obtain a stable product, the entire software stack must be reliable. Therefore software testing must be done at each layer of the software stack, starting with the lowest level: the operating system.

Multiple software testing methodologies are in use nowadays, each of them targeting a degree of test cases coverage and test writing complexity. Alongside the well known unit testing method, another functional methodology named property-based testing has gained ground among software developers. It uses the concept of “tests as

specification”, in which tests are written to cover most of the specification.

Writing a large number of tests for the same specification implies a sizable effort from the developers. Property-based testing mitigates this by automatically generating the input and creating general and abstract tests known as *properties*. Those can be similar to unit tests, except for the way input is generated and output is validated.

This paper presents an user space framework named QC that is based on an open source basic implementation of a property-based testing framework implemented by Pennebaker (2012). Although the well-known related released frameworks are written in functional programming languages, QC is written in C due to the VMXL4 native environment support.

