

Google Summer of Code 2017

The FreeBSD Project

Smoke testing of all base utilities

Introduction

Overview

Smoke testing is a set of light tests which are done for checking basic functionalities of a software to ascertain if the crucial functions work correctly. It precedes the actual in-depth testing process. Smoke testing proves to be very effective in cases when changes done in a software are to be verified if testable for running the actual tests. For e.g. the entire testing process may be rendered useless if the tests are being run on a faulty or bad build environment. The purpose of this project is to develop such smoke-testing suite of all base utilities which can then be integrated with FreeBSD. Once integrated, the tool will also facilitate further development of tests.

About [kyua](#)¹

[kyua](#) is a testing framework for infrastructure software, originally designed to equip BSD-based operating systems with a test suite.

Potential Mentor

Brooks Davis (brooks@freebsd.org)

¹ <https://github.com/jmmv/kyua>

² <https://wiki.freebsd.org/TestSuite>

The problem

FreeBSD currently has a set of tests² under `src/tests` which are run using the `kyua` framework. These tests need to be first installed individually before they can be used for testing. This proves to be problematic in cases when direct testing of some newly installed or updated utility has to be performed. It makes testing changes to utilities and libraries difficult as one would like to perform tests (e.g. to ensure a proper build environment) before proceeding for installation.

Project Abstract

This project aims to develop a new test infrastructure and automation tool along with basic smoke tests to verify if all the base utilities in FreeBSD are linked properly. The testing framework will ease the process of writing test cases which will be run in a completely automated and developer friendly manner without any prior installation.

Implementation Details

The following steps will be performed in order to understand the problem at hand and the approaches which can be used -

- Note how testing has been traditionally done in FreeBSD.
 - Study the `kyua` framework and the approaches currently being used for testing.
 - Study the current problems faced as a developer or tester when writing new test cases, and the problems faced as a release engineer or tester when executing existing test cases.
-

Test Plan

The file [functional_test.c](#) is a simple test file which checks whether the `ls` program is properly linked by running trivial commands. The options supported by `ls` are stored in `short_options[]` and `long_options[]` present in [functional_test.h](#).

The smoke tests can be run using the following commands -

```
>> make
>> ./functional_test <utility> --<option1> --<option2> ...
```

It should be noted that finally the tests will be automated with appropriately passed options. The above commands will not have to be run for testing individual programs.

The test file uses `getopt()` and `getopt_long()` for testing the validity of the passed options. If a valid option is passed, the command `<utility> --<option(i)>` is executed. In case the command fails to execute for a valid option, this will imply that the utility under test is not properly linked.

Populating `short_options[]` and `long_options[]`

`short_options[]` and `long_options[]` need to be initially populated with a few supported options for all the base utilities. This can be done by using either one or both of the following available approaches -

- Parse man pages for each utility to get the supported options.
- Pass an unsupported option to the utility. This might generate a usage message which can then be parsed. The unsupported option will be chosen experimentally.

An automation script will be written which will populate `short_options[]` and `long_options[]` by following the above mentioned approaches and will generate relevant test files.

The initial version of the generated test scripts will only test trivial functionalities.

Although the above mentioned test plan does not check the entire state of application under test, it suffices for the initial purposes of smoke testing.

The test plan for the project will be kept updated at the following location :
<https://github.com/shivrai/smoketestsuite#test-plan>

Expected Results

The final goals which are expected to be completed by the end of the project are -

- An automation tool which can be used for running relevant subsets of the available tests.
- Set of test cases which verify the basic functionality of all the base utilities.
- Generation of a final report from the collected results clearly depicting all the tests which succeeded, failed, skipped or timed-out. The report will be generated in a presentation-independent format which can then be later converted to a user-friendly format.

At the end of the project, all the base utilities will contain relevant tests in their respective directories located under `src/bin`. For e.g. `src/bin/ls/tests`.

Optional (would-be-nice) components -

- Basic tests for verifying if the libraries are properly linked.
-

Deliverables

Deliverables for first evaluation

- Since the number of base utilities is large, manual scripting of tests for each of them will be cumbersome. An automation script will be made which will collect information of the relevant base utilities for producing trivial test scripts initially. The approaches used for this purpose are mentioned in the test plan.

The initially generated scripts will run trivial tests such as `tool --help` and `tool --version` to verify that the program under test is linked properly.

- Test the basic functionality of a subset of base utilities.

Deliverables for second evaluation

- Expand the set of tests to include more functional tests.
- Support for generation of a report depicting important details of the test execution in a presentation-independent format.
- Utility for converting the generated test report into a user-friendly format.

Deliverables for third evaluation

- An automation tool for running relevant subsets of the available tests.
 - Test the basic functionality of all the existing base utilities.
 - An attempt will be made to add smoke tests for verifying whether libraries are properly linked.
-

Project Schedule

| Start | End | Task |
|---------|---------|--|
| 5 May | 30 May | Community Bonding Period Get familiar with the intricacies of smoke testing. Communicate with other testers and release engineers in the community and get their opinions on the plan of action. |
| 30 May | | Start of coding |
| 30 May | 12 June | Start work on writing the automation script for generating trivial smoke tests. |
| 13 June | 25 June | Add smoke tests for a subset of base utilities. |
| 26 June | 30 June | First Evaluations |
| 1 July | 12 July | Start extending the functional tests for each of the base utilities. Start work on writing an automation tool for running relevant subsets of the test scripts. |
| 13 July | 17 July | Add support for generating a report which depicts important details of the test execution in a presentation-independent format. |
| 18 July | 23 July | Add a utility for converting the generated test report into a user-friendly format. |
| 24 July | 28 July | Second Evaluations |
| 29 July | 9 Aug | Make an attempt to add smoke tests for verifying whether libraries are properly linked. |
| 10 Aug | 20 Aug | Add smoke tests for all base utilities. |
| 21 Aug | 29 Aug | Code review and final evaluations Add final touches to the documentation. |

The Code

The source code will be kept updated at : <https://github.com/shivrai/smoketestsuite>

Biography

I am a junior at the Indian Institute of Technology Kanpur, pursuing Bachelor of Science in the department of Mathematics and Scientific Computing.

My interests are in the field of Computer Networks, Operating Systems, Computer Systems Security, Web-Development, Game Theory, Algorithms and Formal Verification.

I am a Unix and Linux enthusiast, and have been using FreeBSD since a year and a half.

Past involvement with The FreeBSD Project

My previous contribution to the FreeBSD community -

TCP/IP Regression Testsuite (Google Summer of Code 2016)

Used : [packetdrill](#), shell scripting, knowledge of Computer Networks

Github link : <https://github.com/shivrai/TCP-IP-Regression-TestSuite>

FreeBSD wiki: <https://wiki.freebsd.org/SummerOfCode2016/TCP-IP-RegressionTestSuite>

Contributions

As a part of preparation for this project after communication with mentor, I wrote a [basic smoke test](#) for the `ls` utility demonstrating the general workflow and test plan which will be followed for further continuing on writing tests.

Availability / Working Hours

My summer vacation will begin from May 4 and end on July 25. During the vacation I will be online on IRC and will be available for work all time during the day when I'm awake. I will continue to be available most of the time after semester begins on July 31.

Contact

Name Shivansh Rai
E-mail shivansh@freebsd.org
Phone +91 7755047792
IRC shivansh on efnet
zeeb on freenode
Blog <https://shivrai.github.io/>
Address C-205
Hall of Residence 10
Indian Institute of Technology
Kanpur, Uttar Pradesh
India - 208016