

ISPRAS Report

March 29, 2008 - April 29, 2008

Abstract

This document presents top-level monthly report on ISPRAS results in the LSB Infrastructure program (which is run jointly with the Linux Foundation).

Revision History

Date	Version	Description
2008-04-30	0.1	Initial version

Contents

ISPRAS ACTIVITIES OVERVIEW.....	1
ACHIEVEMENTS FOR THE PERIOD.....	2
LSB DATABASE.....	2
LSB NAVIGATOR.....	2
LSB DTK MANAGER.....	2
LSB ATK MANAGER.....	3
LSB ECLIPSE PLUG-IN.....	3
LSB CERTIFICATION SYSTEM.....	3
DEEP TESTING.....	4
NORMAL TESTING.....	4
<i>T2C Infrastructure / Technology.....</i>	<i>4</i>
<i>Test Development.....</i>	<i>4</i>
<i>ISPRAS New Normal Tests Summary.....</i>	<i>5</i>
<i>Code Coverage Analysis.....</i>	<i>5</i>
SHALLOW-NORMAL TESTING.....	5
SHALLOW TESTING.....	6
COMMUNITY TASKS.....	6

ISPRAS Activities Overview

The current ISPRAS scope in the LSB Infrastructure Program includes the following areas:

1. Development, maintenance and maturing of **LSB informational systems**:
 - a) The main **LSB Database** (both schema and data) - 71 tables with over 25 million records in total.
 - b) Over 40 **Database Scripts** (including libtodb2 tool for importing libraries).
 - c) Linux developer portal - **LSB Navigator** (<http://linux-foundation.org/navigator/>).
 - d) **LSB Certification System** (<http://linux-foundation.org/lsb-cert/>).
2. Development, maintenance and maturing of **automated test execution and result analysis frameworks**:
 - a) **LSB ATK Manager** for testing and analyzing *applications*.
 - b) **LSB DTK Manager** for testing and analyzing *distributions*.
3. Development, maintenance and maturing of **automated test development frameworks** for different cost & value grades:
 - a) **UniTESK** for deep testing.
 - b) **T2C** for normal testing.
 - c) **Azov** for shallow testing.
4. Development of **new tests** for LSB interfaces in various quality grades:
 - a) Deep tests.
 - b) Normal tests.
 - c) Shallow tests.
5. **Analytical and community collaboration** tasks:
 - a) When developing tests we inspect corresponding specifications and Linux implementations (both manually and by tests) - we analyze, additionally annotate and publish all found issues at http://linuxtesting.org/results/impl_reports and http://linuxtesting.org/results/std_reports, and then cooperate with corresponding authors to make specifications and upstream components finally get fixed.
 - b) We are actively involved in discussing / problem solving of LSB / Linux issues of various kinds in mailing lists / irc.
 - c) We promote and advance LSB and Linux/Open Source in general through participating in Russian government official working groups (to define federal policies), speaking at conferences, publishing papers, organizing events.

Achievements for the period

Detailed list of completed tasks can be extracted from the weekly reports at the <http://www.linux-foundation.org/en/ENG:Status> page (see ISPRAS section at the bottom). The sections below highlight only most important / higher level things.

LSB Database

1. DB Infrastructure (schema, scripts, general issues):
 - a) Released **LSB DB & Scripts Update 1.3** with new way of storing community data, multi-version headers and lots of data corrections in GL/X11 headers and c++ area.
 - b) **3** new tables added to schema and populated with proper data based on a research (InterfaceVote, ApprovedLibrary and ApplInterpreter).
 - c) Multi-LSB-version headers implemented (the main step for multi-version lsbcc).
 - d) Application data collection tools improved to analyze interpreters used by applications.
 - e) libchk, elfchk and stub libraries generation speedup - up to 10 times faster now.
2. LSB Database Data:
 - a) Collected and uploaded data for **3** new distributions and **80** applications (including **50** proprietary). We now have more than **1000** applications in the database.
 - b) Fixed OpenGL DB data. GL headers can now be generated from the database.
 - c) Performed research analysis and prepared data for libstdc++ uplift.
3. LSB Bugzilla Activity:
 - a) **17** new bugs identified and filed.
 - b) Patches for **28** bugs created.
 - c) Investigations conducted and suggestions submitted for **6** bugs.

LSB Navigator

1. **2** new subsections added in the AppStats - "Applications Representatives" and "Interpreters Usage Frequency".
2. "Test Coverage" pages speed up - the main page is about 2 times faster now.
3. Implemented "Community Voting" for interfaces.
4. Class home pages extended with application usage data.

LSB DTK Manager

1. Released [LSB DTK Manager 1.5.7](#):
 - a) added support for Qt3-AZOV test kit (shallow tests);
 - b) bug fixes in test suites: Core (bug [#1978](#) related), Xts5 and Apache.
2. Further development of **DTK-2** (based on a new architecture to address fixes and improvements almost impossible in the current DTK architecture):
 - a) a lot of changes in report generator that are aimed to support manual tests and make report generator more flexible and customizable;
 - b) fixed several bugs in the ptysell tool;
 - c) new algorithm of multi-journal comparison and multi-comparison of whole reports;

LSB ATK Manager

1. Released [LSB ATK Manager 1.4](#):
 - a) completely redesigned Test Results page;
 - b) added possibility to group problems by category;
 - c) all architecture names are internally standardized;
 - d) **5** bugs fixed (incl. [#152](#), [#2019](#), [#2021](#), [#2022](#)):
2. Further improvement of the Web-UI:
 - a) added "Save As..." to the Upload form to save the package for manual sending via e.g. E-mail;
 - b) added the "Clear Form" button on the "Application Check" Web-UI page.

LSB Eclipse Plug-in

1. Developed 5 new project types:
 - a) "C++" Debug Executable;
 - b) "C" Release Shared Library;
 - c) "C" Debug Shared Library;
 - d) "C++" Release Shared Library;
 - e) "C++" Debug Shared Library.
2. Implemented opening Web-UI of the ATK Manager right within Eclipse.
3. Cmd-UI of the ATK Manager is now used for testing application under development.

LSB Certification System

1. **5** new major features:
 - a) Added invoice & payment workflow support (status, notifications, warnings).
 - b) Added option for claiming *member* status for company at registration (with further admin notification for review and approval).
 - c) Dramatically improved internal mechanism of validation of user submitted data.
 - d) Added list of all (active and completed) certifications in admin mode (per company and overall).
 - e) Basic support of changing fee schedule with time – supporting multiple fee schedules depending on the certification date (will be improved further).
2. **6** minor improvements:
 - a) Corrected sorting of architecture names.
 - b) More hyper-links to interlink all the pieces in admin mode.
 - c) Returning to the right page after attachment deletion.
 - d) Improved functions for paging of long lists.
 - e) Few optimizations in the database schema.
3. **16** bugs found at QA and fixed.

Deep Testing

1. Tests for **3** interfaces upgraded to the deep level (io.fstream.lock subsystem).
2. **18** new requirements are now covered by improving test scenarios.
3. "signal" formal model was developed.
4. Implemented new automated report with work progress and nightly builds summary.

Normal Testing

T2C Infrastructure / Technology

1. Added support for generating catalog of requirements from several source html files at once.
2. Enhanced T2C code generator to support sets of parameter values. Some minor improvements are still needed here but this new functionality already works correctly in common use cases. More thorough testing of this functionality is now in progress.
3. T2C tools can now generate a standalone version of a whole test suite, these tests do not use TETWare tools at all.
4. Fixed: multiline text of a requirement was loaded incorrectly from the catalog before execution of tests.

Test Development

GTK

1. Previously developed tests for libgdk-pixbuf-2.0 (**71** interface, **373** test cases) and libgobject-2.0 (**314** interfaces, **926** test cases) have been released after another review along with many updates of the tests released before (e.g. for libfontconfig).

OpenGL

1. Tests have been developed for **70** more interfaces from libGL: **4270** test cases added.

Now the status of normal test development for OpenGL is as follows:

Library	Tested interfaces	Number of test cases
libGL	186	7157

X11

1. **197** test cases have been developed for **95** more interfaces from libX11.
2. To mark up the requirements for these interfaces in the documentation, html version of its **10** chapters has been prepared (reformatted) from various sources.

Now the status of test development for X11 is as follows:

Library	Tested interfaces	Number of test cases
libX11	131	248

C++

1. **247** test cases have been prepared for **150** interfaces (mostly methods of locale and several facet classes).

Now the status of test development for libstdcxx is as follows:

Library	Tested interfaces	Number of test cases
libstdcxx	224	445

ISPRAS New Normal Tests Summary

Now the "Desktop-T2C" test suite developed by ISPRAS (numbers are in total since the beginning) contains over **11500** new "normal" test cases for more than **1900** interfaces from the following libraries:

Library	Tested interfaces	Number of test cases
libglib-2.0	828	1938
libgthread-2.0	2	2
libgmodule-2.0	8	28
libatk-2.0	222	574
libfontconfig	160	314
libgdk-pixbuf-2.0	71	373
libgobject-2.0	314	926
libGL	186	7157
libX11	131	248
Total	1922	11560

Additionally, "C++-T2C" test suite contains about **450** test cases for more than **220** interfaces:

Library	Tested interfaces	Number of test cases
libstdcxx	224	445

Code Coverage Analysis

Analysis was performed to find out how much of the code of **4** libraries from glib2 package (version 2.16.3) is covered (executed) when the tests from Desktop-T2C test suite run. The results are presented in the table below. Note that non-LSB interfaces from these libraries were not taken into account.

To our knowledge of common industry standards, the results are very good.

Library	Lines of code (total)	Executed lines	Code coverage
libglib-2.0	16263	12203	75.0 %
libgthread-2.0	211	149	70.6 %
libgmodule-2.0	270	211	78.1 %
libgobject-2.0	7000	5605	80.1 %
Total	23744	18168	76.5 %

Shallow-Normal Testing

Based on the latest experience we need to introduce an intermediate level of testing quality – *shallow-normal*. The reason is that the high speed of shallow testing development using Azov technology is not achievable for all types of libraries. Sometimes it is wiser to use T2C technology to develop the "shallow-normal" tests when we need to speed up – the full T2C technology is used but with relaxed requirements to the number of things to be checked. Especially this is important when we create both normal and shallow tests inside the same library. Azov and T2C are completely different technologies and different skills are needed. But in the "shallow-normal" approach the development for particular library stays within the same group of people (so we reuse library and technology expertise). Also, "shallow-normal" tests are easy materials to upgrade to normal when we have time in future.

Currently we use the "shallow-normal" approach in OpenGL testing (note that we develop both normal and shallow-normal tests there). For the recent month we have developed new "shallow-normal" tests for **111** libGL interfaces (and 70 normal ones as described above).

Shallow Testing

1. Created RPMs with qt3 shallow-tests for all 7 architectures and integrated them into local DTK-manager.
2. More corrections in data for Qt4 in database: added missed parameters and return types for interfaces from Qt4 library, added about 1000 interfaces and 700 types for Qt4 library, uploaded pure virtual methods.
3. Report system update was integrated into Azov framework.
4. Debug option in makefiles for building shallow-tests was added (in t2c format).
5. New shallow tests developed for Qt4 library (**3346** done from 11282):
 - a) libQtCore library (**1589** done from 2127).
 - b) libQtGui library (**1397** done from 7702).
 - c) libQtNetwork library (**45** done from 417).
 - d) libQtSql library (**317** done from 377).

Community Tasks

1. Usual participation in the lsb-* lists.
2. Found, analyzed and published **9** confirmed problems in Linux upstream components and specifications (see http://linuxtesting.org/results/impl_reports and http://linuxtesting.org/results/std_reports).
3. Jointly with *Ministry of Telecommunications and IT* developed a list of most important open-source projects to recommend for government support (LSB is among them).
4. Prepared and submitted **2** articles for the SYRCoSE 2008 conference (<http://syrcose.ispras.ru/program.html>) - "Linux Distributions and Applications Analysis During Linux Standard Base Development" and "Extension of Interface Signature Descriptions for Automatic Test Generation". Accepted by program committee.
5. Talked at "R&D and Application of High Technologies" conference (St-Petersburg) on "*Using Open Source Projects for University Education*".
6. Participated in "4th Russian Open Source Forum" at Interop 2008. Spread Linux/LSB ideas.
7. Participated in LF Collaboration Summit 2008 (Austin, TX).