



ISP-RAS Activities in Open-Source Software

Arutyun Avetisyan

Institute for System Programming

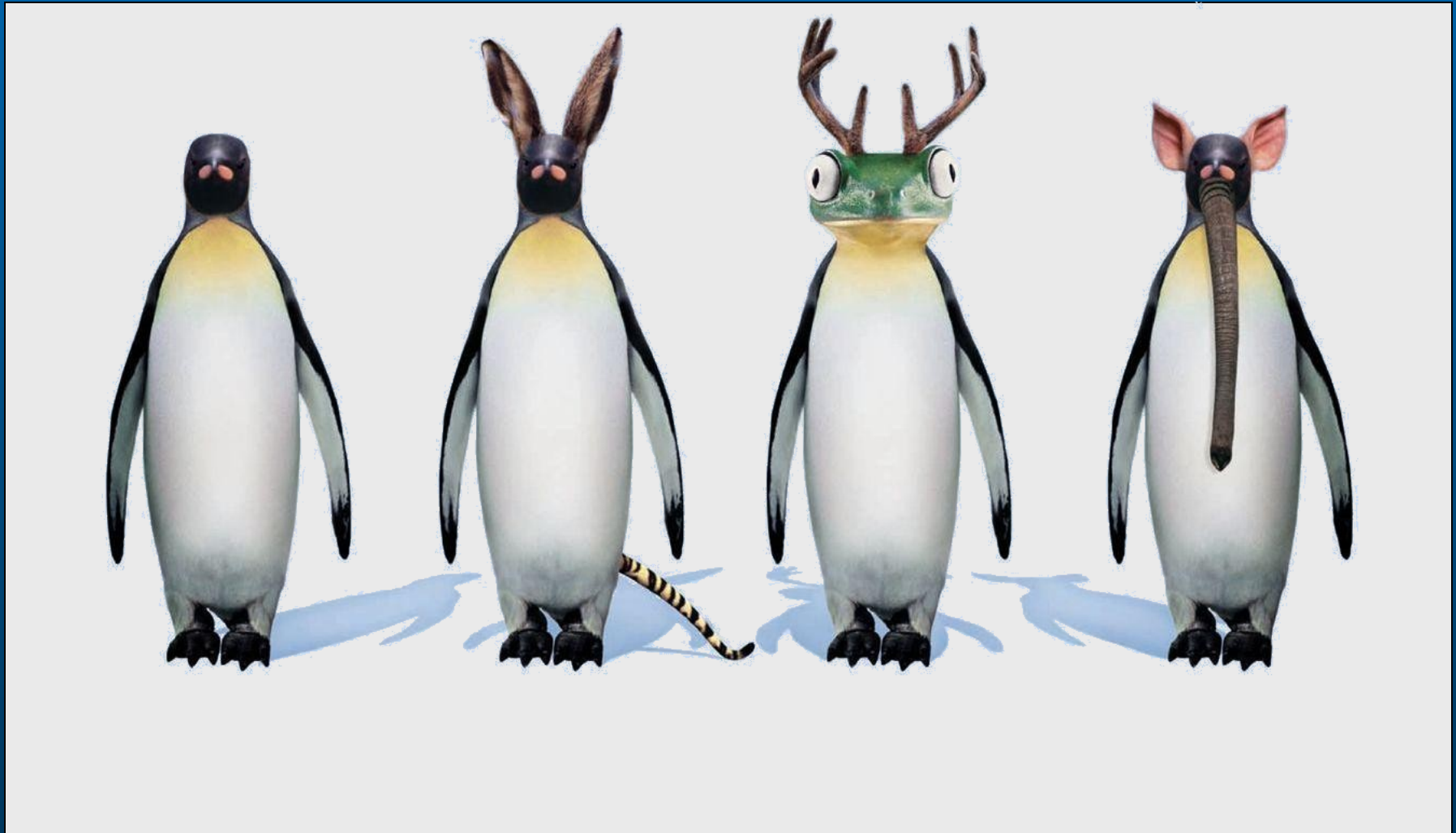
Russian Academy of Sciences

What is Linux?



- There are hundreds of Linux distributions but there is no “single Linux” everybody adopts.
- Linux is not just the kernel.
- Linux distribution is a collection of various upstream components such as the kernel, libraries and applications.
- Linux should mean a platform.

Linux Distributions



Why we need Linux standardization?

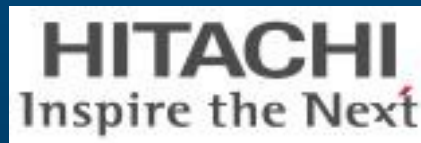


- Standardization defines the single Platform – a common functionality subset of all the Linux distributions.
- Any standard application works on all the Linuxes and any Linux supports all the standard applications.
- Lower development and support costs.
- More applications due to standard interoperability.

Linux Standard Base (LSB)



- The key Linux standard
- Developed by Free Standards Group (currently Linux Foundation) since 2000
- LSB 3.1 U1 (current version) – March 2007
- Has become ISO/IEC 23360 standard in 2005
- Linux Foundation Platinum Members:



LSB 3.1 Overview



- Specifies behavior of binary interfaces of the main system libraries (about 30000 functions)
- Defines directory structure and main system commands as well as ELF and RPM formats
- Refers to existing standards:
 - ◆ Single UNIX Specification - POSIX
 - ◆ ISO C99
 - ◆ Upstream documentation
 - ◆ ...
- Itanium (IA64) architecture is one of the primary targets for architecture specific standardization

Linux Verification Center at ISP RAS



- Based on **ISP RAS** background in operating systems and advanced testing technologies **Russian Federal Agency for Science and Innovations** established the Linux Verification Center in 2005
- The mission is to help advance Linux by:
 - ◆ Development of Linux test suites
 - ◆ Strengthening Linux open standards
 - ◆ Dissemination of academic and industrial expertise
- The Center is in close cooperation with:
 - ◆ The Linux Foundation (<http://linux-foundation.org/>)
 - ◆ Austin Common Standards Revision Group (<http://opengroup.org/austin/>)

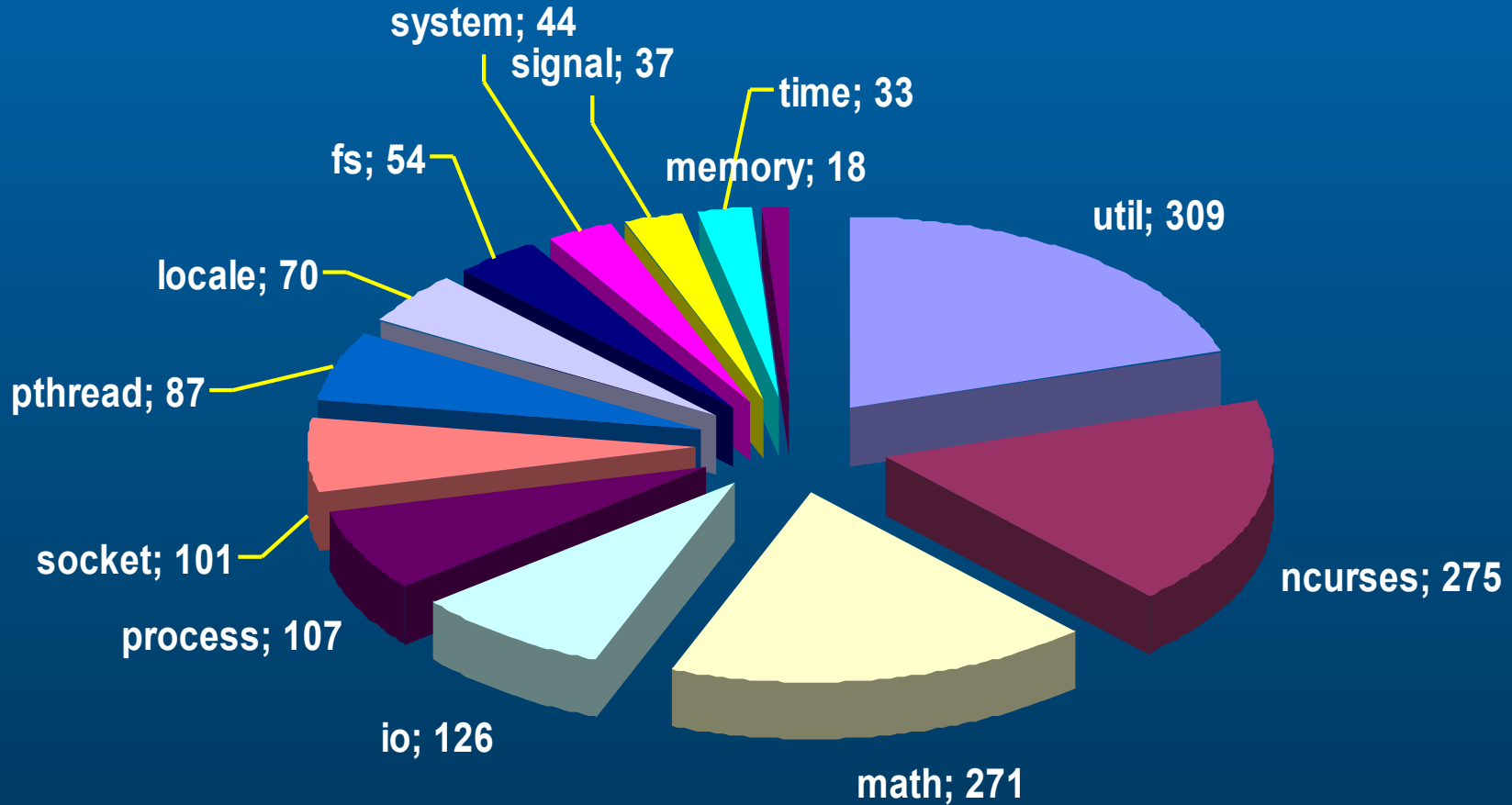
OLVER Project



Open Linux VERification:

- Development of an open source test suite for functional testing of Linux system interfaces against LSB Core 3.1 standard's requirements (1532 system interfaces).

LSB Core Interfaces

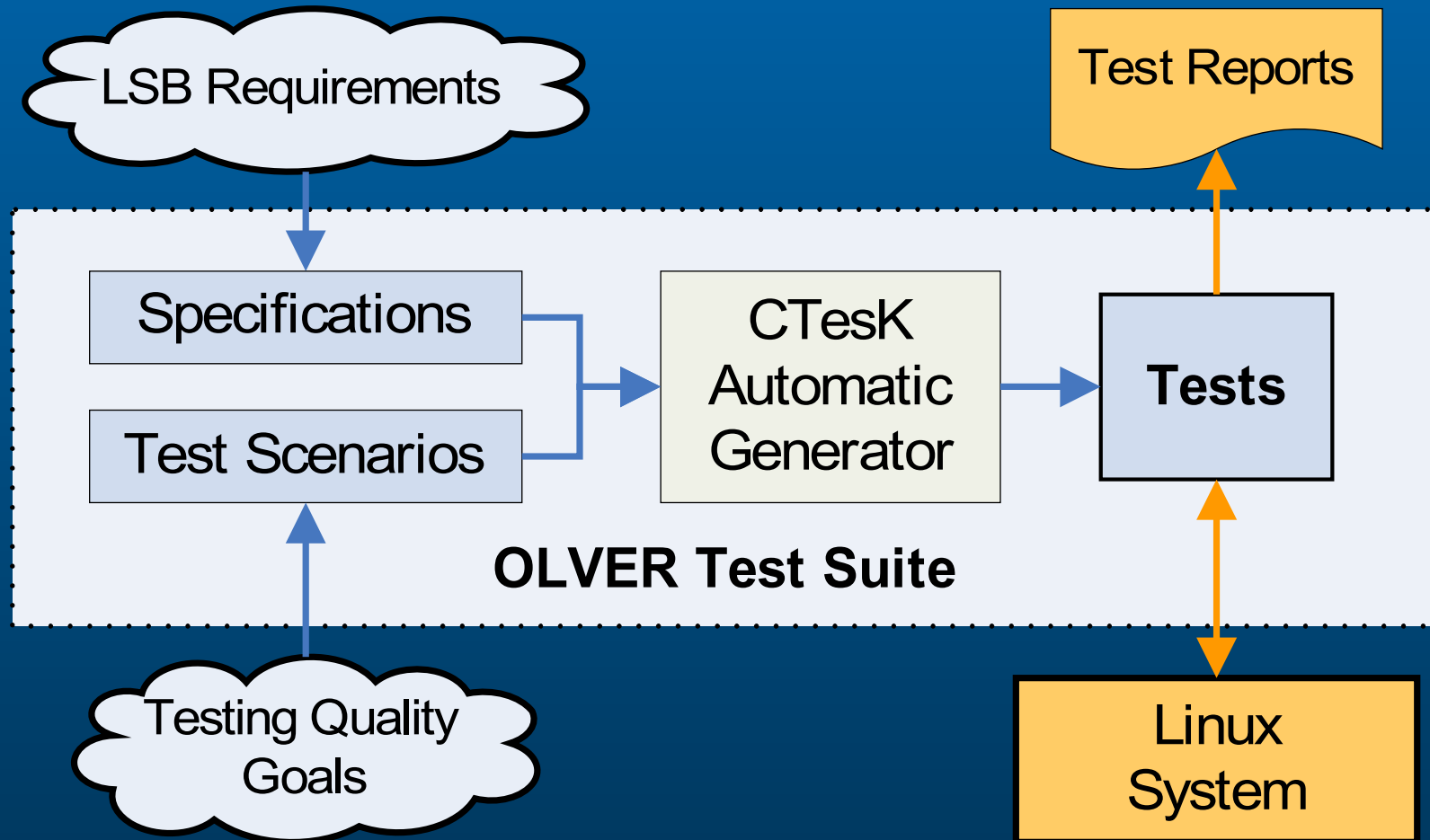


OLVER Technological Steps



- Analyze LSB Core 3.1 standard's text and extract atomic requirements (assertions)
- Formalize the assertions into SeC (Specification extension of C) specifications
- Developing test scenarios in SeC language
- Automatic test generation based on the formal specifications and test scenarios (UniTESK technology)

OLVER Tests Development



OLVER Results



- The LSB standard's text has been analyzed for all the **1532 generic interfaces**. Online requirements catalog has been created.
- Discovered and submitted **70 problem reports** for the text of LSB and POSIX standards.
- **Tests** created for **1270** (good quality) + **260** (basic quality) interfaces
- More than **130 bugs** found in the modern distributions.
- All the results are open source and available at <http://linuxtesting.org/>

LSB Core Test Coverage



Subsystem	Interfaces	GLIBC	LSB	LTP	OLVER 1.0
fs	54	39	33	42	47
io	128	78	92	73	62
locale	70	56	60	19	61
math	271	259	31	39	271
memory	18	12	14	17	18
ncurses	275	0	59	3	272
process	106	61	77	81	86
pthread	87	79	66	80	87
signal	37	25	24	30	27
socket	101	34	2	56	47
system	44	20	15	23	26
time	33	23	26	27	33
util	308	187	144	76	233
Bcero	1532	873	643	566	1270

LSB Infrastructure Program



- Started in September 2006
- Customer: The Linux Foundation (formerly Free Standards Group)
- Top level goal: provide uniform infrastructure for contributing to LSB and ensure LSB delivers real value in solving compatibility problems.

ISPRAS

**THE LINUX
FOUNDATION**

**LinuxTesting
.org**

LSB Infrastructure Program



- Improve test coverage to have 100% interfaces tested (quality should vary depending on the interface importance).
- Improve quality of the specifications (standard's text).
- Develop an integrated LSB Database and populate necessary data.
- Develop new integrated software infrastructure systems:
 - ◆ LSB Navigator – collaboration portal
 - ◆ LSB Test Execution Framework
 - ◆ LSB Certification System
- Ensure linkage between all the key elements – distributions, applications, upstream components, tests and LSB specification.

Defect Detection Environment



- ❑ A tool for Security Vulnerabilities and Critical Errors Detection in C/C++ program source code allows to discover the following defect types:
buffer overflow, access beyond the object bounds, format string vulnerability, null pointer dereference, use after free, double free, memory leak
- ❑ Good soundness and low false positives rate
- ❑ Advanced data-flow based algorithms of analysis (including interprocedural level)
- ❑ Original methods of backward analyses for detecting source reasons for each warning

Contacts



- Arutyun Avetisyan
arut@ispras.ru
- Vladimir Rubanov
vrub@ispras.ru
- Institute for System
Programming of the RAS
<http://ispras.ru>
- Linux Verification Center
<http://linuxtesting.org>
info-lvc@linuxtesting.org

