Informational System to Support Development and Usage of Linux Interface Standards

Denis Silakov

Institute for System Programming, RAS
http://ispras.ru/
Linux Verification Center
http://linuxtesting.org/

SYRCoSE 2010. 1-2 June, 2010, Nizhny Novgorod
Linux Ecosystem

- **System components ~ 5,000**
  - (kernel, libraries, utilities, …)
  - developed independently
  - “release early, release often”

- **Distributions ~ 500**
  - based on “upstream” system components
  - add their own patches
  - a set of selected applications

- **Applications ~ 10,000**
  - want to run on many distributions
## Components in Distributions

### Distributions released November, 2009

*Component versions and number of functions exported by component libraries*

<table>
<thead>
<tr>
<th>Component</th>
<th>Mandriva 2010</th>
<th>Fedora 12</th>
<th>openSUSE 11.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLIBC</td>
<td>2.10 2275 functions</td>
<td>2.11 2283 functions</td>
<td>2.10 2275 functions</td>
</tr>
<tr>
<td>GTK</td>
<td>2.18.3 4518 functions</td>
<td>2.18.3 4915 functions</td>
<td>2.18.1 4915 functions</td>
</tr>
<tr>
<td>ALSA</td>
<td>1.0.21 1623 functions</td>
<td>1.0.21 1623 functions</td>
<td>1.0.21 1609 functions</td>
</tr>
</tbody>
</table>
Application Portability

- Thoroughly test in every system
  - by application developers
  - by maintainers in distributions
  - requires significant resources

- Give source code to users
  - also necessary for distribution maintainers
  - not everyone wants to share source code

- Follow standards
  - API – recompile for every system
  - ABI – use binary executables and libraries 'as is'
  - development of a standard can be a challenge
Target Area

- A Linux distribution: \textbf{1.500} libs, \textbf{1.000.000} functions
- Applications use from 10 to 10.000 functions
- POSIX: 1.500 functions, LSB: \textbf{40.000} functions
- How to select what to standardize?
- Profiles?
Accompanying Products

- Test suite
- Build environment
- Sample implementation
- ...
- Should be kept synchronized
Developing a Standard

Constant monitoring of the Linux Ecosystem
- Interfaces provided by leading distributions
- Interfaces used by popular applications
- Can be automated

Selection of candidates for next Standard version
- Formal rules based on the monitoring results, e.g.: interfaces present in all systems released after 2008
- Can be automated, too

Finalization of list of candidates, manual actions
- Create documentation
- Develop tests
- ...
Workflow
We consider **binary** applications only

- Structural – can be analyzed statically
  
  *e.g.*, synopsis of functions in header file

- Semantic – require interface invocation to be analyzed
  
  *e.g.*, function behavior

Analysis of structural properties is enough to check if application can be **launched** in distribution
Temporal Relationship Model (TRM)

- Extension of 'usual' relational model
- Life period for every element: $[T_s..T_e]$, possible values for $T_s .. T_e$ – standard versions + NULL
- Can be served by relational DBMS, but improvements requiered in tools that work with database
Time Intervals

- Discrete time, small set of possible values
- Dependencies by time between connected items

<table>
<thead>
<tr>
<th>Function</th>
<th>Assigned to Header</th>
<th>Appeared in</th>
<th>Withdrawn in</th>
</tr>
</thead>
<tbody>
<tr>
<td>gets</td>
<td>stdio.h</td>
<td>1.0</td>
<td>1.2</td>
</tr>
<tr>
<td>fgets</td>
<td>stdio.h</td>
<td>1.0</td>
<td>NULL</td>
</tr>
<tr>
<td>puts</td>
<td>stdio.h</td>
<td>2.0</td>
<td>NULL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Header</th>
<th>Appeared in</th>
<th>Withdrawn in</th>
</tr>
</thead>
<tbody>
<tr>
<td>stdio.h</td>
<td>1.0</td>
<td>NULL</td>
</tr>
</tbody>
</table>
LSB Database

Standardized Elements
- Information about all LSB versions
- LSB 4.0: \(~40,000\) functions from 57 libraries

Linux Ecosystem
- 250 distributions
- \(~1400\) applications

Auxilliary
- Test coverage
- URLs to online documentation for functions
- ...
- LSB specification text
- Some test suites for distributions
- Linux Application Checker
- LSB Build Environment
A Web system upon the LSB DB

- Browsing the database
- Statistical queries
- Analytical queries (decision making support)
  - Interface usage in applications
  - Interface presence in distributions
  - ...

LSB Navigator
• LSB Infrastructure Project
http://ispras.linuxfoundation.org
• LSB at the Linux Development Network
http://ldn.linuxfoundation.org/lsb
• Denis Silakov
silakov@ispras.ru