Flock 2016

Ensuring ABI stability in Fedora

Presented by,
Dodji Seketeli <dodji@fedoraproject.org>
Sinny Kumari <sinnykumari@fedorapoject.org>

Agenda

- What we mean by “ABI”?
- ABI compatibility
- Fedora ABI compatibility verification tooling
- ABI change reports examples
- Possible improvements

What we mean by “Application Binary Interface”

- Context:
  - A binary “E” which uses code from binary “L”
    - “E” can be:
      - Executable
      - Shared library
    - “L” can be:
      - Shared library
      - Dynamically loaded module

What we mean by “Application Binary Interface”

- **At execution time, “E” expects properties from “L”**
  - Format, architecture
  - Presence of certain symbols
  - Specific layout of data
  - Etc …
- Those properties are structural, not behavioral

What we mean by “Application Binary Interface”

- These loose and unwritten structural expectations between “E” & “L” are the ABI.
- We talk specifically about the ABI of a binary:
  - The set of symbols it defines and exports
  - The layout of data expected by these symbols
  - Etc …

ABI changes are inevitable

- Shared libraries need to evolve
  - Bug fixes
  - Features
- New functions and global variables will be added
- Types of existing functions are going to change

Need to detect harmful changes

- Only ABI-incompatible changes are harmful
- For instance:
  - Removal of existing functions
  - Incompatible data layout changes
    - Removal of a struct data member
    - Insertion of a new data member in the middle of a struct
    - Removal of a function parameter
- ABI-compatible changes are fine
- Need to detect ABI-incompatible changes
  - By looking at the binaries only
  - As soon as possible
- Many ABI changes need a human to determine compatibility
  - We are using the “diff” paradigm to represent an ABI change
  - So people can review “ABI diffs” rather than “source code diffs”

Fedora ABI compatibility verification tooling

- For each koji package update build:
  - ABI-compare the new package to the old one
  - Send a message to package maintainer with the “ABI diff”
- Some automatic categorization of ABI changes
  - Incompatible ABI changes flagged as FAILED
  - Gray area changes flagged as NEED INSPECTION
  - Identical ABIs flagged as PASSED

Fedora ABI compatibility verification tooling

- Based on Taskotron
  - Taskotron task named 'abicheck'
  - Compares ABI of new package against previous version tagged as stable
  - Uses libabigail 'abipkgdiff' command line tool for ABI comparison
    - Package maintainers can use libabigail command line tools offline
      - “fedabipkgdiff”
    - Upstream hackers too!
      - “abipkgdiff”
      - “abidiff”
    - Everyone should review the ABI changes of their shared library before releasing!
- Limitations
  - C/C++ shared libraries
  - Runs on a sub-set of critpath packages

ABI change report example

- Real example from https://taskotron.fedoraproject.org/artifacts/all/6ee5e57e-525d-11e6-ae46-525400120b80/task_output/gpgme-1.6.0-3.fc23.log

ABI change report example (1/3)

* **ABI changes found between gpgme-1.4.3-6.fc23.x86_64.rpm and gpgme-1.6.0-3.fc23.x86_64.rpm. ABI comparison took 3.18 second(s). Please review them**.

========== changes of 'libgpgme-pthread.so.11.11.0 ==========

Functions changes summary: 0 Removed, 1 Changed (115 filtered out), 7 Added functions

Variables changes summary: 0 Removed, 0 Changed, 0 Added variable

ABI change report example (2/3)

7 Added functions:

'function const char* gpgme_get_dirinfo(const char*)' {gpgme_get_dirinfo@@GPGME_1.1}

'function int gpgme_get_offline(gpgme_ctx_t)' {gpgme_get_offline@@GPGME_1.1}

'function void gpgme_get_status_cb(gpgme_ctx_t, gpgme_status_cb_t*, void**)' {gpgme_get_status_cb@@GPGME_1.1}

(...)

ABI change report example (3/3)

* **1 function with some indirect sub-type change:**

[C]'function gpgme_error_t gpgme_cancel(gpgme_ctx_t)' at gpgme.c:194:1 has some indirect sub-type changes:

parameter 1 of type 'typedef gpgme_ctx_t' has sub-type changes:

underlying type 'gpgme_context*' changed:

in pointed to type 'struct gpgme_context' at **context.h:76:1:**

type size changed from 1664 to 1792 bits

3 data member insertions:

'unsigned int gpgme_context::offline', at offset 416 (in bits) at context.h:102:1

'gpgme_status_cb_t gpgme_context::status_cb', at offset 1216 (in bits) at context.h:139:1

'void* gpgme_context::status_cb_value', at offset 1280 (in bits) at context.h:140:1

Improvement directions

- Taskotron / Fedora infra level
  - More memory and processors for a given task
  - Handle suppression specifications for tests in general
- Task-abicheck level
  - Gradually increase the set of ABI-verified packages
  - Take package API (devel sub-package) into account
- Abipkgdiff / libabigail level
  - Decrease memory usage for pathological cases
  - Support more C/C++ language constructs
  - Better ABI change categorization
  - More web friendly reporting
  - Dedicated ABI changes tracking web service

Questions?

- https://fedoraproject.org/wiki/Taskotron/Tasks/abicheck
- https://fedoraproject.org/wiki/ABICompatibilityDefinitions
- https://taskotron.fedoraproject.org/resultsdb/results?testcase_name=dist.abicheck
- https://www.sourceware.org/libabigail/wiki

Thanks!