Standards for Device-side Test Artifacts

Tim Bird

Principal Software Engineer

Sony Electronics

Abstract

This session will involve a discussion around a proposal for standards for device-side test artifacts. Currently there are no standards (that the author is aware of) for where tests should be placed in a device under test, or how test frameworks should discover, interact with, and collect results from test artifacts.

Tim will propose adding some new directories to the FileSystem Hierarchy Standard to specify that:

- test code and data should go under /usr/test
- a test wrapper function, called "{testname}-run" should be placed in /usr/test/bin
- test output should be placed in /var/test (or maybe /var/log/test) with name "{testname}-output-{datestamp}.{appropriate-extension}"
- a user account called "test“, and group called “test” should be created, with well known pid 88 and gid 88
    - The directories and files above should be owned by 'test.test'

This would allow end users and automated tools to find and easily execute any tests that are packaged with a system. It also designates a place in the filesystem where tests can be placed. Having separate locations for test artifacts allows for different mounting or storage decisions for those locations in the filesystem. This could be beneficial since tests might not be part of production releases, or test artifacts might only be applied to a device temporarily.

This is intended to be a discussion among automated testing and distribution developers, to see if this is something useful going forward, and to plan next steps.

SONY

Agenda

- Proposal
- What does it solve?
- Existing landscape
- Issues
- How to proceed?
- Discussion

SONY

Proposal (in abstract)

- Declare a standard for:
  - How to detect tests on a Linux system
  - How to execute a test
  - How to find test results

SONY

Details

- Create a new standard directory: /usr/test
  - with subdirectory: /usr/test/bin
    - bin directory used for tests, wrappers and testing-specific utilities
  - with subdirectory for each suite: /usr/test/{testname}
- Standard name for a test or test wrapper:
  - /usr/test/bin/{testname}-run
- Create new standard test user and group:
  - user ‘test’, uid=88; group ‘test’, gid=88
  - Used by default for test execution (or ‘root’, depending on testing policy)
  - artifacts in /usr/test would be owned by ‘test.test’
- Test output stored by default in directory: /var/test
  - or maybe /var/log/test?
  - test results placed in a file named: {testname}-{timestamp}.{ext}

SONY

What does this solve?

- Allows test frameworks to find, enumerate, execute and retrieve results from installed tests
  - Tests can be added or removed, and system will automatically detect currently available tests
- Allows for test artifacts to be stored separately from “production” binaries and data
  - Test artifacts, particularly in embedded, may be too big to stay resident in production devices (ex: LTP)
  - Can choose a test storage policy appropriate for product or usage scenario

SONY

What does this solve? (cont.)

- Predefined ‘test’ user and group allow for more consistent test execution
  - Tests and frameworks can rely on a pre-existing non-root user account
    - One less configuration item for automated test setup
  - Permission checks can be more consistent
- /usr/test/bin (and maybe /usr/test/lib) can serve as a place to store common test helper tools and libraries
  - Can build up a base of useful libraries and utilities, in one place
- /var/test allows users and tools to find results in a common location
  - Storage policy for results can be managed separately from other system elements (logs, email, runtime data, etc.)

SONY

Vision

- Create an ecosystem where:
  - End-users can easily execute tests and send results
  - Test frameworks can automatically find and execute tests, and collect results

Existing landscape

- LTP
  - Linux
  - Posix
- Yocto ptest
- Debian Autopkgtest
- Fuego

Existing landscape

System	Default test directory	Default user	Default executable
LTP (Linux)	/opt/ltp/testcases/bin or /opt/ltp/runtest	root or {user}	{testcase} or specified in runtest file
LTP (Posix)	/opt/ltp/conformance/{type}/{category}	root or {user}	{testcase}.run-test
Yocto ptest	/usr/lib/{pkg-name}/ptest	as executed by test user	run-ptest (eg. /usr/lib/{pkg-name}/ptest/run-ptest)
Debian Autopkgtest	debian/tests/{testname}	as executed by test user	specified in debian/test/control file
Fuego	/home/fuego/fuego.{testname}	configurable	specified in fuego_test.sh
Proposal	/usr/test/bin /usr/test/{testname}	root or ‘test’	{testname}-run

Issues

- Security
  - Test frameworks may automatically run items they discover in /usr/test/bin
    - Could introduce a new location for malware injection
    - However, having /usr/test separate from rest of system may allow easier fencing of test executables
- Storage types
  - Users (or vendors) need to determine storage policy for test artifacts
    - read-write vs. read-only, persistent vs. temporarily installed, local vs. remote, same partition or different partition from production filesystem, etc.
- Results output format:
  - Is outside the scope of a filesystem/user account standard
  - However, would be nice to have consistent output format per test executable
    - e.g. Have two different executables for LTP Posix and LTP Linux tests

SONY

How to proceed

- Find out if there's any interest...
- Find out if there are any existing standards
  - That supercede this, or that this should extend or contemplate
- Discuss additional items
  - Additional directories: /usr/test/lib, /usr/test/man?
  - Additional conventions: standard command line args for output formats?
    - e.g. –json = json output, --kernelci= kernelci output
- Add to Filesystem Hierarchy Standard
- Introduce and promote to distribution vendors and build systems
  - Anyone who packages tests, or who packages software that includes tests
  - Developers who write tests?

SONY

Poll – Find interest level

- What is your interest level in a Linux directory standard for tests?
  - A – Yeah!! = I'm interested, and willing to adopt this standard in my packages, distribution, or framework
  - B – OK = I'm interested, but I probably won't do anything myself
  - C – Wait and see = If this gets traction, I'm willing to use it
  - D – Needs more work = Potentially interesting, but needs changes
  - E – Nah = Not interested / This isn't a useful idea

SONY

Discussion

SONY

Existing landscape details

SONY

Linux Test Project (LTP)

- Default location:
  - /opt/ltp/testcases/bin (for Linux tests)
  - /opt/ltp/runtest (for Linux test groups)
  - /opt/ltp/conformance/[interfaces|behavior]/{category} (for Posix tests)
- Default user/group: as executed by test admin
- Default executable:
  - Linux: specified in runtest file
  - Posix: {testcase}.run-test
- Helper scripts and utilities: t0, run-tests.sh ltp-pan, and others

Yocto ptest

- Default location: /usr/lib/{pkg-name}/ptest
- Default user/group: <unknown> (as executed by test admin?)
- Default executable: run-ptest
- Helper scripts: ptest-runner2

SONY

Debian autopkgtest

- Default location: debian/tests/{testname}
- Default user/group: <unknown> (as executed by test admin?)
- Default executable: specified in debian/tests/control file
- Helper scripts: /tmp/autopkgtest-reboot
- Resources:
  - https://people.debian.org/~mpitt/autopkgtest/README.package-tests.html

Fuego

- Default location: /home/fuego/fuego.{testname}
  - prefix is configurable (can be /home/test or /root)
- Default user/group: is configurable, commonly ‘fuego’, ‘test’, or ‘root’
- Default executable: specified in fuego_test.sh
- Helper scripts
  - /home/fuego/fuego.{testname}/fuego_board_function_lib.sh