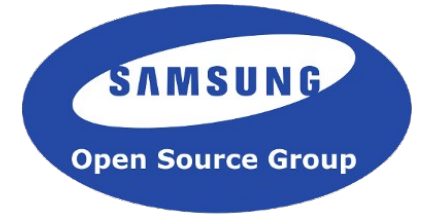


Linux Kernel Selftest Framework BoFs - Quality Control for New Releases

ELC 2015

Shuah Khan
Samsung Open Source Group
shuahkh@osg.samsung.com

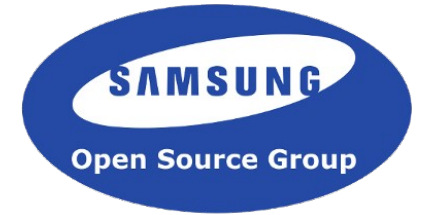
Abstract



Kselftest is an effort to enable a developer-focused unit test framework in the kernel to ensure the quality of new kernel releases. The project is making steady progress on the plans and goals that emerged from the Kernel summit Kselftest session.

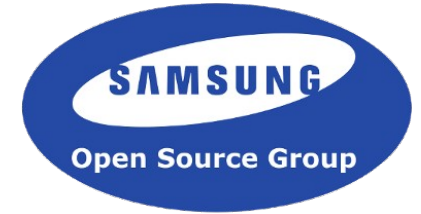
The development efforts since the Kernel summit discussion have been focused on adding more tests, common framework for test reports, and the install target feature. This talk will discuss the work done so far as well as future plans in detail. In addition, it will cover how to get involved through adding tests, evolving the framework to install on targets, and enabling running tests and quality of reporting test results.

Agenda



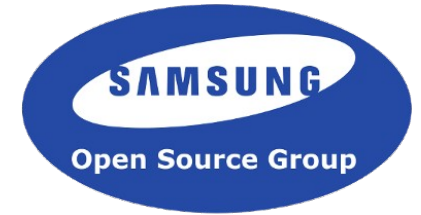
- Kselftest – what is it?
- Kernel summit and beyond ...
- Where are the tests?
- Kselftest git
- How to run selftests?
- How to run a subset of selftests?
- Full range testing ...
- How to install selftests?
- Run tests from install location ...
- Contributing new tests ...
- Next Steps

Kselftest – what is it?

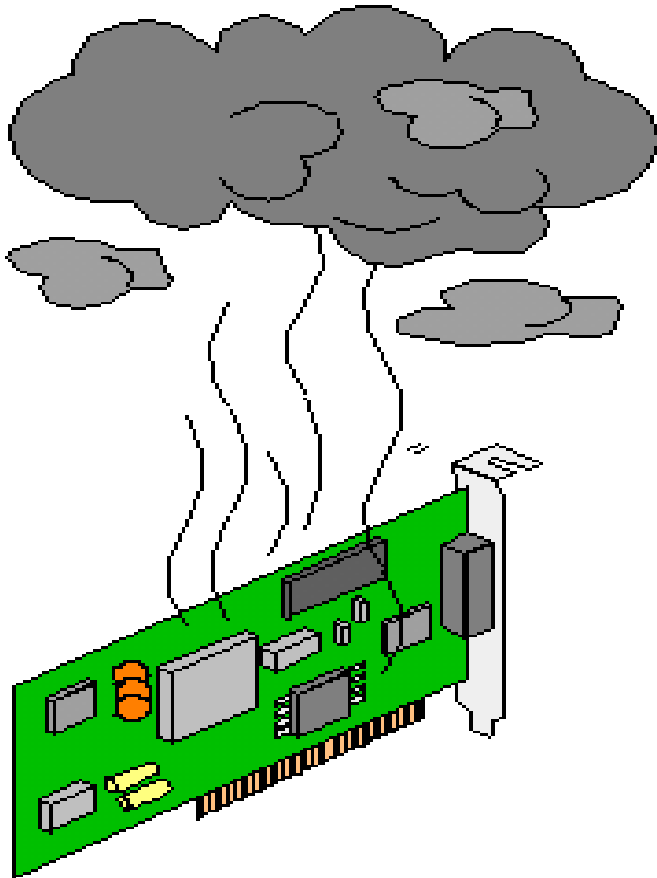


- A battery of developer regression tests:
 - Regression test an existing or a new feature
- Leverages existing tests
- Ability to quickly execute tests (current target is under 20 minutes)
- Keep tests updated as new features get added to the kernel
- Encourage test submissions
- Test source code in the kernel source tree

Kselftest – goals



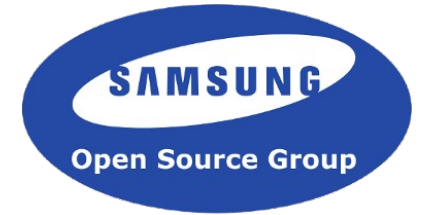
- Enable kernel developers to run regression tests - primary
- Enable ability to install and run tests on a target systems large and small (qemu)
- Enable ability to generate package to install and run tests on target large and small (qemu)
- Ability to run different groups of tests: quick/long/destructive



Tests

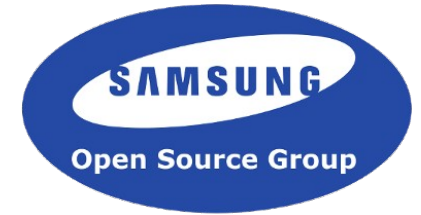
_____	✓
_____	✗

Kernel summit and beyond ---



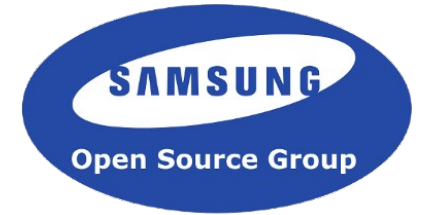
- Features added in 3.17:
 - kselftest make target to main Makefile
 - enables running tests from the kernel main Makefile
 - enables kernel developers to run regression tests

Kernel summit and beyond ---



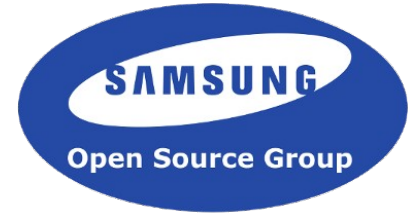
- Features added in 3.18:
 - ftrace tests (Masami Hiramatsu)
 - execveat() system call test – a good example of test going in with the system call.
 - new powerpc tests
- Features added in 3.19:
 - system size test (Tim Bird)
 - Documentation to kernel Documentation directory
 - Kselftest framework for reporting results
 - Some tests updated to use the kselftest framework
- 4.0 includes several fixes.

Kernel summit and beyond ---

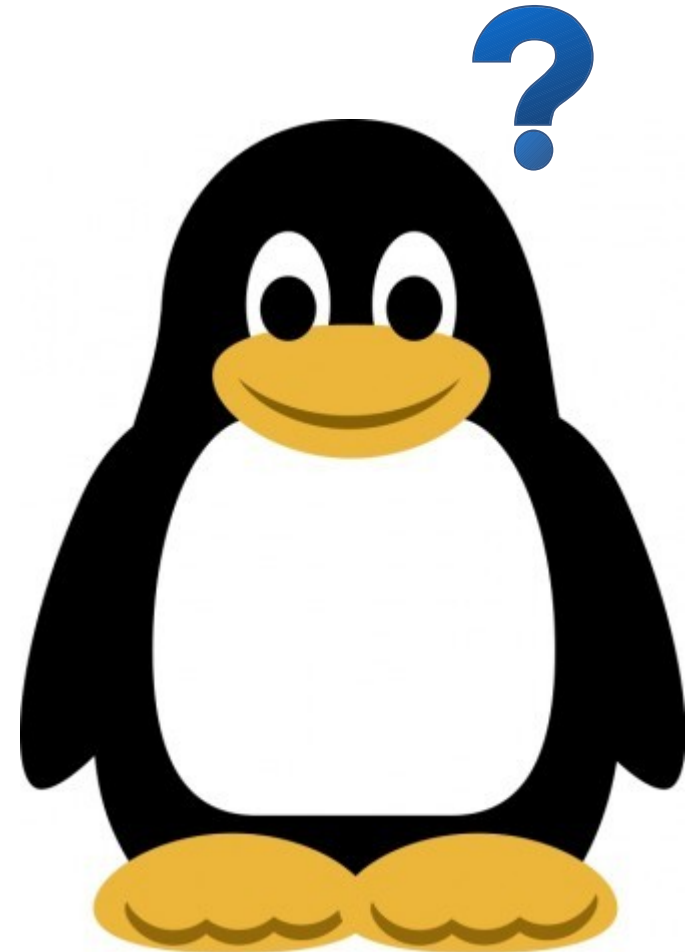


- Features queued for 4.1 in linux-kselftest next:
 - kselftest Install target (Michael Ellerman)
 - time tests (John Stultz, Linaro)
 - kselftest install and packaging tools (shuah Khan)
 - rtc tests (Prarit Bhargava, Red Hat)

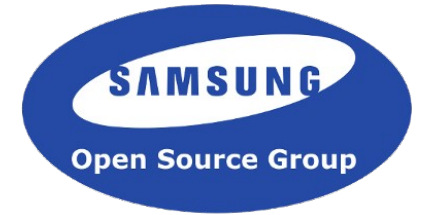
Where are the tests?



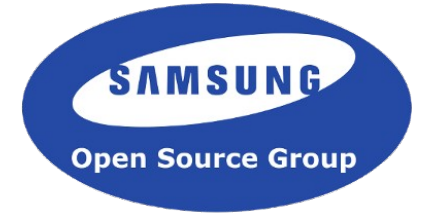
- tools/testing/selftests



Kselftest git



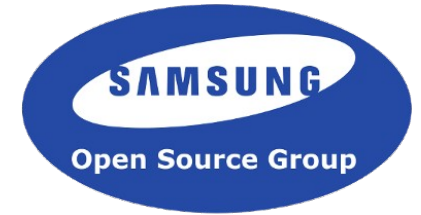
- Kselftest git hosted on kernel.org
- Branches fixes, next, and devel are maintained.
- fixes: contains fixes to current rc's
- next: contains content for upcoming merge window.
- devel: contains experimental patches
- <https://git.kernel.org/cgit/linux/kernel/git/shuah/linux-kselftest.git/>



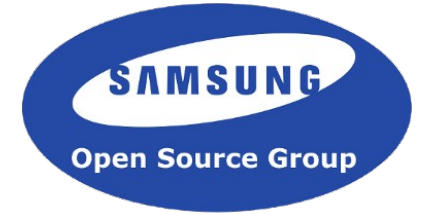
How to run selftests?

- To build and run tests: (from the top level kernel source directory
 - make kselftest
- To build tests:
 - make -C tools/testing/selftests
- To run the tests:
 - make -C tools/testing/selftests run_tests
- Note that some tests require root privileges to run.

How to run a subset of selftests?

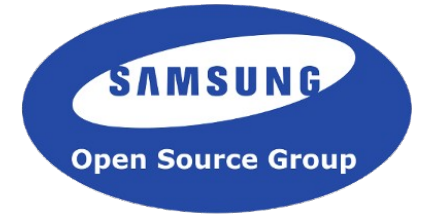


- Run tests for a single or multiple sub-systems using TARGETS variable
- Run tests for multiple sub-systems:
 - make TARGETS="size timers" kselftest
- Run tests for a single sub-system:
 - make TARGETS="size" kselftest



Full range testing

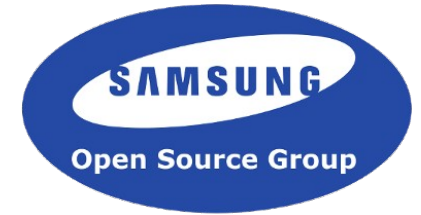
- Some tests e.g: hotplug support full and default (reduced) range modes.
- In full range, test will run the test on all hot-pluggable cpu and memory modules.
- `make -C tools/testing/selftests run_hotplug`



How to install selftests?

- Run kselftest install tool in tools/testing/selftests
 - `cd tools/testing/selftests`
 - `./kselftest_install.sh [install_location]`
- Default install location:
 - `tools/testing/selftests/kselftest`
- Specify install location:
 - `./kselftest_install.sh /tmp`

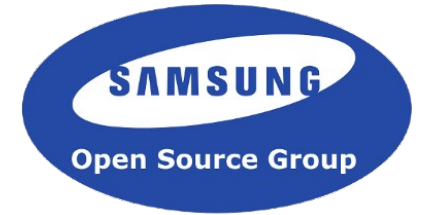
How long does it take to run tests?



make kselftest - completes under 12 minutes.

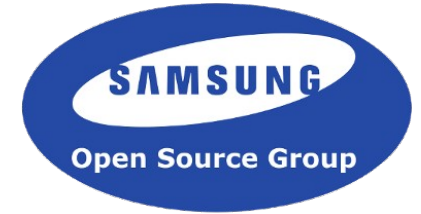
```
real 11m50.499s
user 3m25.979s
sys 5m45.433s
```


Run tests from install location ...



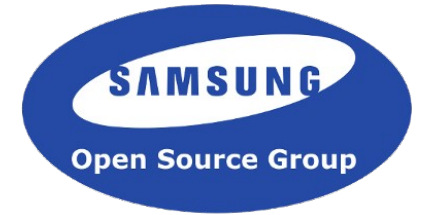
- Kselftest install creates `run_kselftest.sh`
- Run tests:
 - `cd install_dir`
 - `./run_kselftest.sh`
- Kselftest run from the install location completes in under 12 minutes.

Generate tar archive ...



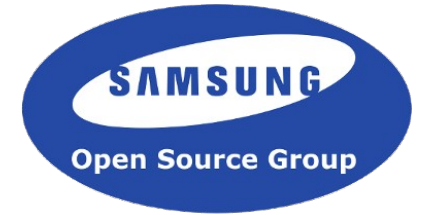
- Run generate tar tool in tools/testing/selftests
 - cd tools/testing/selftests
 - ./gen_kselftest_tar.sh [tar | targz | tarbz2 | tarxz]
- Tool supports uncompressed tar, gz, bz, and xz compression formats.
- Default is .gz

Contributing new tests



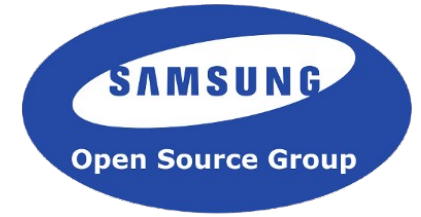
- Report pass and fail conditions.
- When run as non-root, try testing the parts that can be tested without being root.
- Aim for shorter test runs. Don't bloat the full kselftest run times.
- Ensure build doesn't break on any architecture.
- Ensure “make kselftest” doesn't break if feature is unconfigured.
- Ensure test exits gracefully if feature is unconfigured.

Next Steps ...



- Several sub-projects are identified at the Kernel Summit:
 - Standardize the test output format (Grant Likely)
 - Support cross-compilation (Tim Bird, Grant Likely, and others)
 - Support local execution of tests in a virtualized environment (kvm, virtualbox, qemu)
 - Documentation:
 - Docs should go in the kernel Documentation directory – Done
 - <https://kselftest.wiki.kernel.org/> - Done
 - Integration with rest of kernel build system to support standard kernel options:
 - Separate output directory (ie. KBUILD_OUTPUT support)
 - Verbose vs. quiet building (ie V=1 support)
 - Cross-compilation (ie CROSS_COMPILE support)

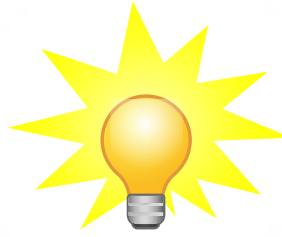
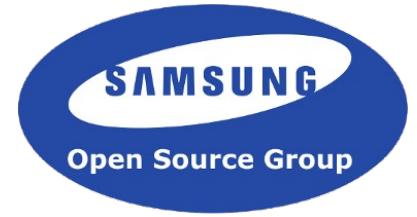
Next Steps ...



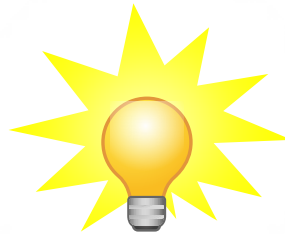
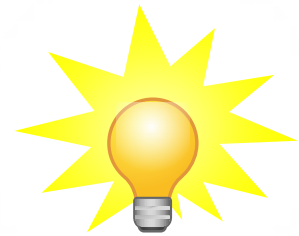
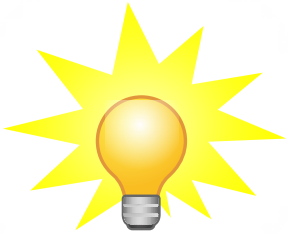
- Desired new tests:
 - System size test – done
 - Time tests – in 4.1
 - Performance tests
 - Basic boot test (get to login prompt)

109 year old Alfie Date from Australia spends his time knitting sweaters for injured penguins!!





Discussion Time!





Thank You!