

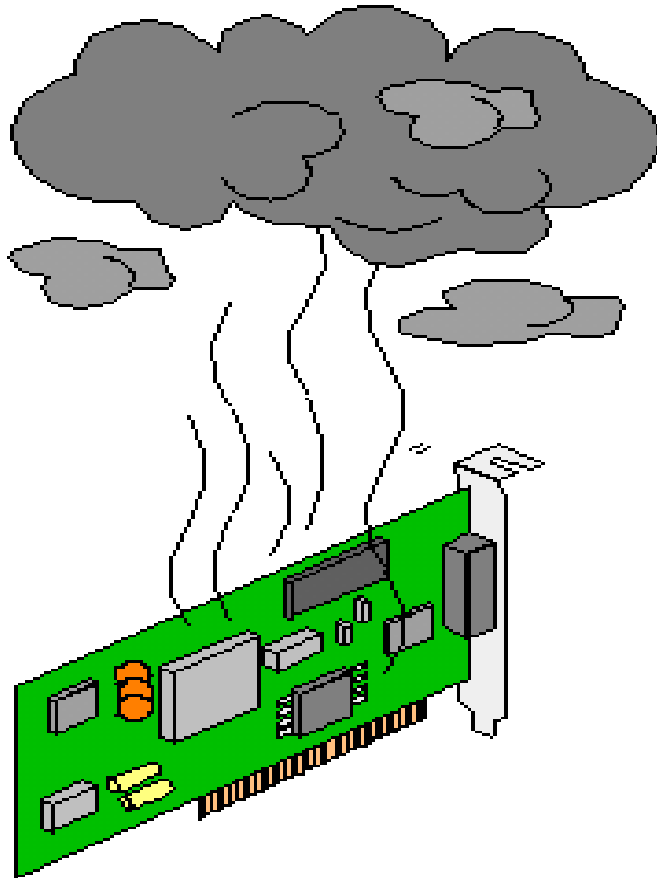
All about Kselftest

Open Source Summit Aug 31 2018

Shuah Khan
Samsung Open Source Group
shuah.kh@samsung.com
shuah@kernel.org
@ShuahKhan

What is Kselftest?





Tests

_____	✓
_____	✗

Kselftest Evolution ...

- Kernel developer/user regression test suite
- Authors: Kernel developers, users
- Users: Kernel developers, users, automated test rings (Kernel CI, 0-day)
- Gets run in stable and mainline release cycles

Kselftest Framework ...

- Common infrastructure for:
 - Building, Running, reporting
- Makefile, lib.mk – abstracts common code for Makefile targets:
 - run_tests, install etc.
- kselftest.h – abstracts TAP13 reporting
- kselftest_harness.h – test harness used in some tests

Kselftest Tools ...

- Install and packaging:
 - `kselftest_install.sh`
 - `gen_kselftest_tar.sh`

Kselftest reporting results ...



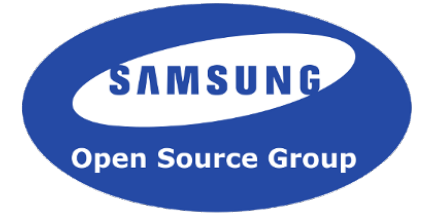
Kselftest results ...

- TAP13 compliant test reporting

TAP13 format ...

- Simple text based interface
- Aids in detecting run-to-run differences
- Makes it easier for external tools to parse the results
- **TAP13 - Test Anything Protocol**

Kselftest use-cases ...



Kselftest from same Linux 4.18 git repo

Linux 4.18 kernel

Kselftest from mainline latest

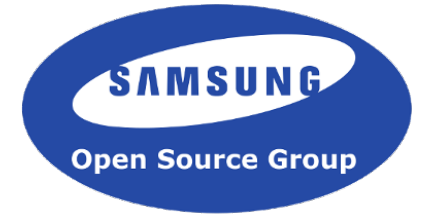
Stable release

Install and run Kselftest on a target

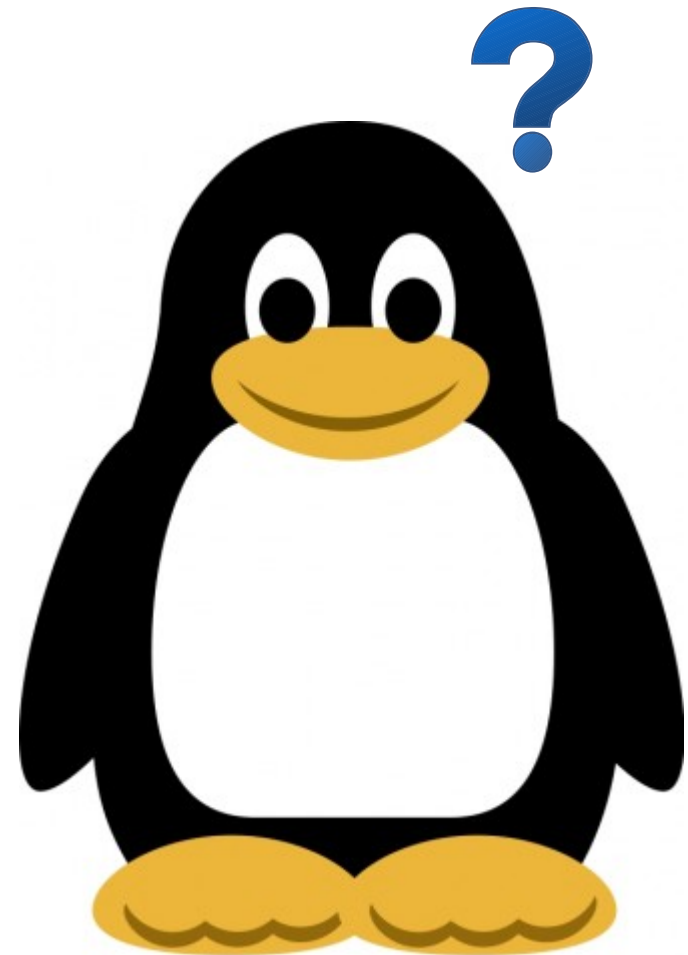
Kselftest source ...



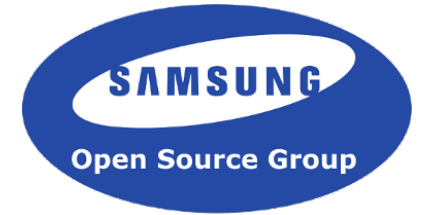
Where are the tests?



- tools/testing/selftests



Kselftest git



- Kselftest git hosted on kernel.org
- Branches:
 - fixes: contains fixes to current rc's
 - next: contains content for upcoming merge window.
 - devel: contains experimental patches
- [linux-kselftest git](#)
- [Patchwork - linux-kselftest](#)

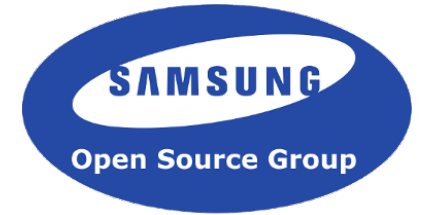
References ...

- ../Documentation/dev-tools/kselftest.rst
- My blogs ...
 - [Kselftest for Linux 4.14 to Add Support For Test Object Relocation](#)
 - [Kselftest for Linux 4.13 to Include TAP13](#)
 - [An Introduction to Testing the Linux Kernel with Kselftest](#)

Building tests ...

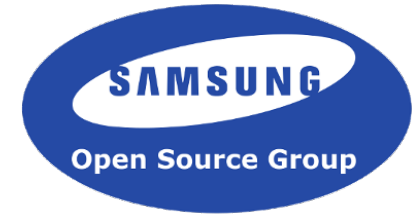


Building tests ...



- `make –silent -C tools/testing/selftests`
 - Builds all tests
- `make –silent -C tools/testing/selftests/timers`
 - Builds timers tests

Building tests – cross compile ...



- `make –silent -C ARCH=arm64
CROSS_COMPILE=aarch64-linux-gnu-
tools/testing/selftests`
 - Builds all tests
- `make –silent -C ARCH=arm64
CROSS_COMPILE=aarch64-linux-gnu-
tools/testing/selftests/timers`
 - Builds timers tests

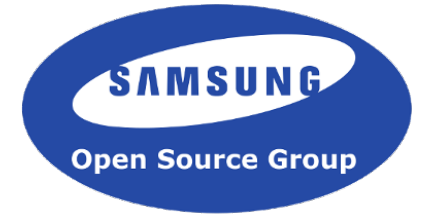
Running tests ...



Running tests ...

- `make –silent kselftest`
 - Kernel Makefile `kselftest` target - Builds and runs all TARGETS in `tools/selftests/Makefile`
- `make –silent -C tools/testing/selftests run_tests`
 - Builds and runs all TARGETS in `tools/selftests/Makefile`
- `make –silent TARGETS=timers kselftest`
 - Kernel Makefile `kselftest` target - Builds and runs all non-destructive tests in `tools/selftests/timers`
- Note that some tests require root privileges to run.

Running tests - TARGETS var ...



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

Running tests – make O=dir ...

- `make –silent O=/tmp/kselftest kselftest`
 - Builds and runs all TARGETS in `tools/selftests/Makefile`
 - Executables are created in `O=/tmp/kselftest/test` directories
- `make –silent TARGETS=timers O=/tmp/kselftest kselftest`
 - Builds and runs all non-destructive tests in `tools/selftests/timers`
 - Executables are created in `O=/tmp/kselftest/timers`

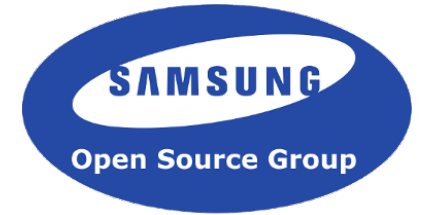
Running individual tests ...

- `make –silent -C tools/testing/timers run_tests`
- `make –silent -C tools/testing/breakpoints run_tests`
- `make –silent -C tools/testing/kcmp run_tests`

Run kcmp test ...

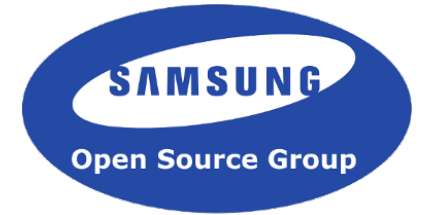
- `make –silent TARGETS=kcmp kselftest`
- `make –silent TARGETS=kcmp O=/tmp/kselftest`
- `make –silent -C tools/testing/kcmp run_tests`

Default vs. stress testing ...



- hotplug (cpu and memory), pstore tests support stress and default modes.
- Kselftest test runs them in default mode.

Stress test runs ...



- run the test on all hot-pluggable cpu and memory modules.
 - `make –silent -C tools/testing/selftests run_hotplug`
- pstore run crash test:
 - `make –silent -C tools/testing/selftests run_pstore_crash`
- timers
 - `make –silent -C tools/testing/selftests/timers run_destructive_tests`

Test results ...



Detailed vs. summary ...

- Kselftest reports detailed results in default mode
 - Includes messages from individual tests.
- Supports summary option for just the pass/fail/skip counts
 - make `–silent summary=1 kselftest`

Code Slide

```
make --silent -C tools/testing/selftests/sync run_tests
TAP version 13
selftests: sync: sync_test
=====
1..0 # Skipped: Run Sync test as root.
not ok 1..1 selftests: sync: sync_test [SKIP]
```

Code Slide

```
make --silent -C tools/testing/selftests/sync/ run_tests summary=1  
TAP version 13  
selftests: sync: sync_test  
=====  
not ok 1..1 selftests: sync: sync_test [SKIP]
```

Code Slide

```
sudo make --silent -C tools/testing/selftests/sync run_tests
TAP version 13
selftests: sync: sync_test
=====
# [RUN] Testing sync framework
ok 1 [RUN] test_alloc_timeline
ok 2 [RUN] test_alloc_fence
ok 3 [RUN] test_alloc_fence_negative
ok 4 [RUN] test_fence_one_timeline_wait
ok 5 [RUN] test_fence_one_timeline_merge
ok 6 [RUN] test_fence_merge_same_fence
ok 7 [RUN] test_fence_multi_timeline_wait
ok 8 [RUN] test_stress_two_threads_shared_timeline
ok 9 [RUN] test_consumer_stress_multi_producer_single_consumer
ok 10 [RUN] test_merge_stress_random_merge
Pass 10 Fail 0 Xfail 0 Xpass 0 Skip 0 Error 0
1..10
ok 1..1 selftests: sync: sync_test [PASS]
```

Code Slide

```
make --silent -C lib/ run_tests
TAP version 13
selftests: lib: printf.sh
=====
printf: module test_printf is not found [SKIP]
not ok 1..1 selftests: lib: printf.sh [SKIP]
selftests: lib: bitmap.sh
=====
bitmap: module test_bitmap is not found [SKIP]
not ok 1..2 selftests: lib: bitmap.sh [SKIP]
selftests: lib: prime_numbers.sh
=====
prime_numbers: module prime_numbers is not found [SKIP]
not ok 1..3 selftests: lib: prime_numbers.sh [SKIP]
```

Installing tests ...



Kselftest install ...

- 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`

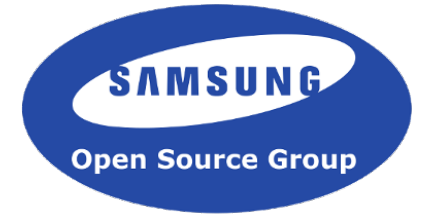
Kselftest install run ...

- Kselftest install creates `run_kselftest.sh`
- Run tests:
 - `cd install_dir`
 - `./run_kselftest.sh`

Kselftest package ...



Generating kselftest tar-ball ...



- 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

Helpful Tips and Hints ...



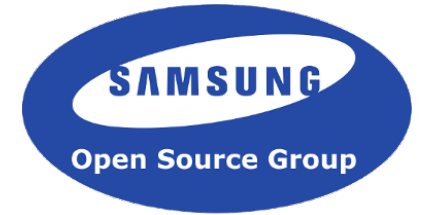
Tips and hints ...

- Use `make –silent` to silence makefile specific output.
- Running as root increases coverage.
- Beware of tests that could be disruptive – might require reboot.
- Running Kselftest from mainline kernel on Stable releases is recommended.

Tips and hints ...

- Running tests from the latest release on stable releases:
 - Works.
 - Offers better regression testing.
- Comparing results from run to run helps detect regressions.
- Save old results and compare with the new rc results.

Contributing new tests



- Report pass and fail conditions.
- When run as non-root, try testing the parts that can be tested without being root.
- Ensure build doesn't break on any architecture.
- Ensure “make kselftest” doesn't break if feature is not supported.
- Ensure test exits gracefully with `ksft_skip` code when test can't run due to unmet dependencies.

What's new in Linux 4.18 ...

- Skip handling ...
 - Kselftest framework reports skipped tests
 - Individual tests detect and return `ksft_skip` to framework
 - Several new tests

Tests as of 4.18-rc1 ...

- android
- bpf
- breakpoints
- capabilities
- cgroup
- cpufreq
- cpu-hotplug
- drivers
- efivarfs
- exec
- filesystems
- firmware
- ftrace
- futex
- gpio
- ia64 (not a run_tests TARGET)

Tests as of 4.18-rc1 ...

- intel_pstate
- ipc
- kcmp
- kmod
- kvm
- lib
- locking
- media_tests
- membarrier
- memfd
- memory-hotplug
- mount
- mqueue
- net
- networking (not a run_tests TARGET)
- nsfs

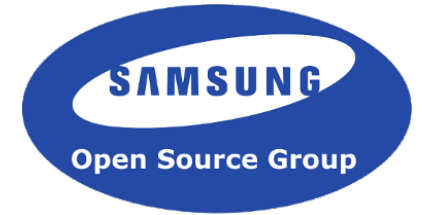
Tests as of 4.18-rc1 ...

- ntb (not a run_tests TARGET)
- powerpc
- prctl
- proc
- pstore
- ptp
- ptrace
- rcutorture (not a run_tests TARGET)
- rseq
- rtc
- seccomp
- sigaltstack
- size
- sparc64
- splice
- static_keys

Tests as of 4.18-rc1 ...

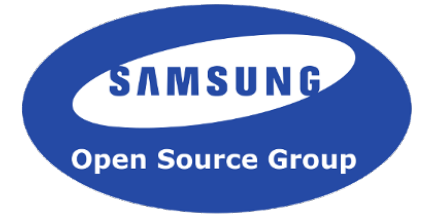
- sync
- sysctl
- tc-testing (not a run_tests TARGET)
- timers
- uevent
- user
- vDSO (not a run_tests TARGET)
- vm
- watchdog (not a run_tests TARGET)
- x86
- zram

Next Steps ...



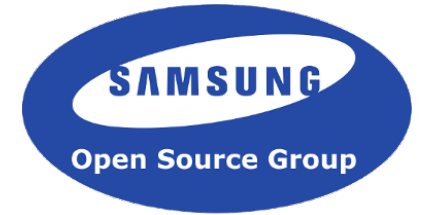
- Run-time documentation
 - What does the test do?
- Improved skip reporting to include reasons:
 - destructive/intrusive/unsupported etc.

Next Steps ...



- Fix problems with O= relative paths don't work
- Add support for pre-processor output from main Makefile (e.g: *.i)
- Consistent headers inclusion across tests

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



Help needed ...



- Review tests
- Run tests as a complement to existing tests
- Contribute new tests (driver area is weak)



Thank You!