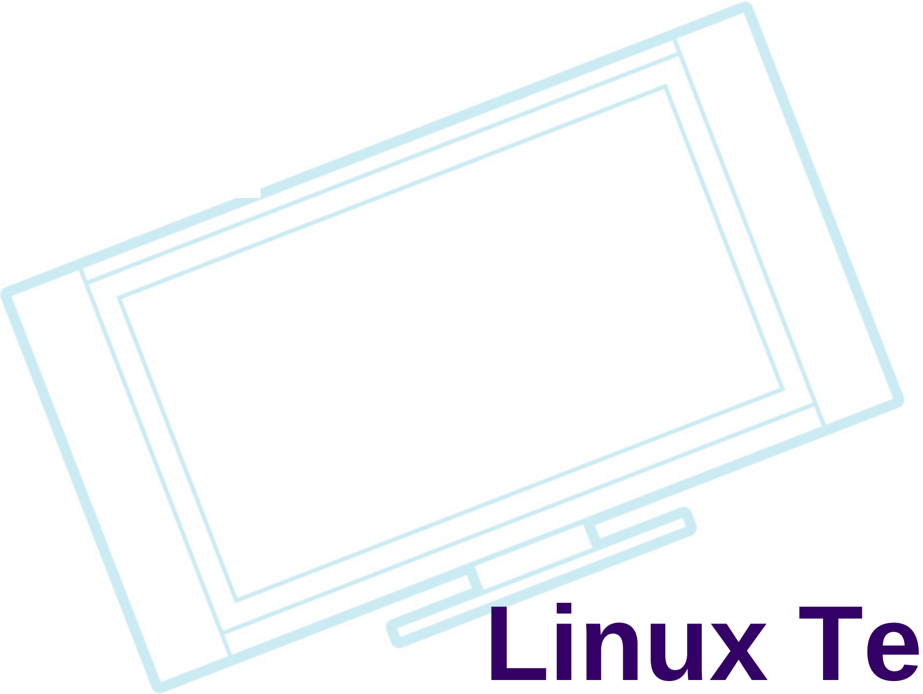
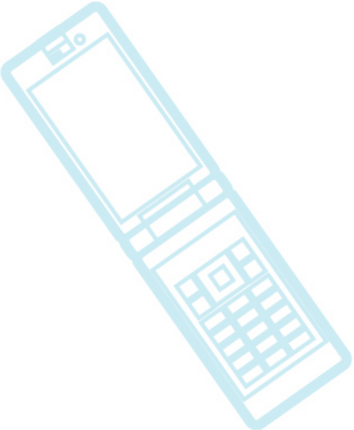
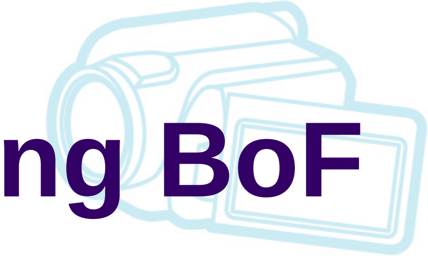
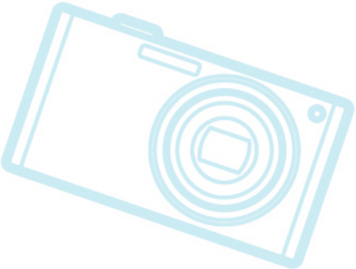
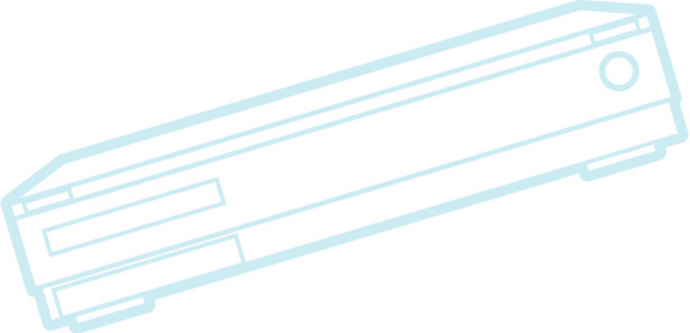






Linux Testing BoF





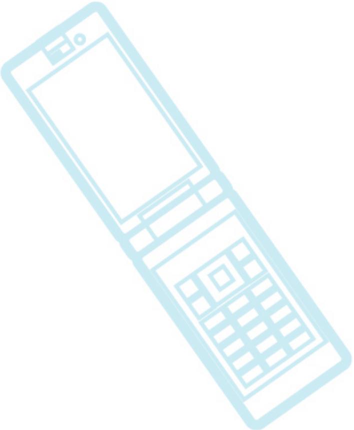
June, 2017

Tim Bird

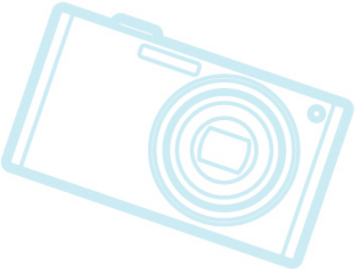
Sony Electronics

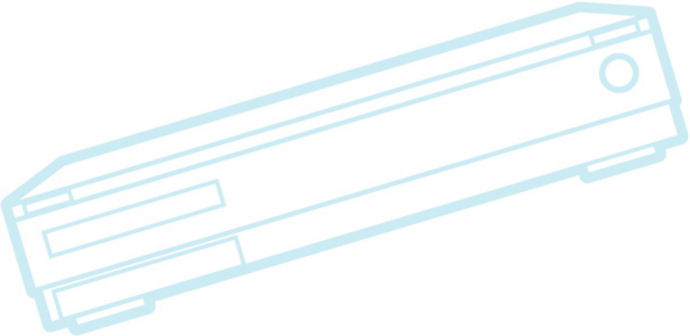
Agenda

- Overview of existing Linux tests
    - With material shamelessly taken from Guenter Roeck!
- Fuego roadmap update
- Discussion of issues



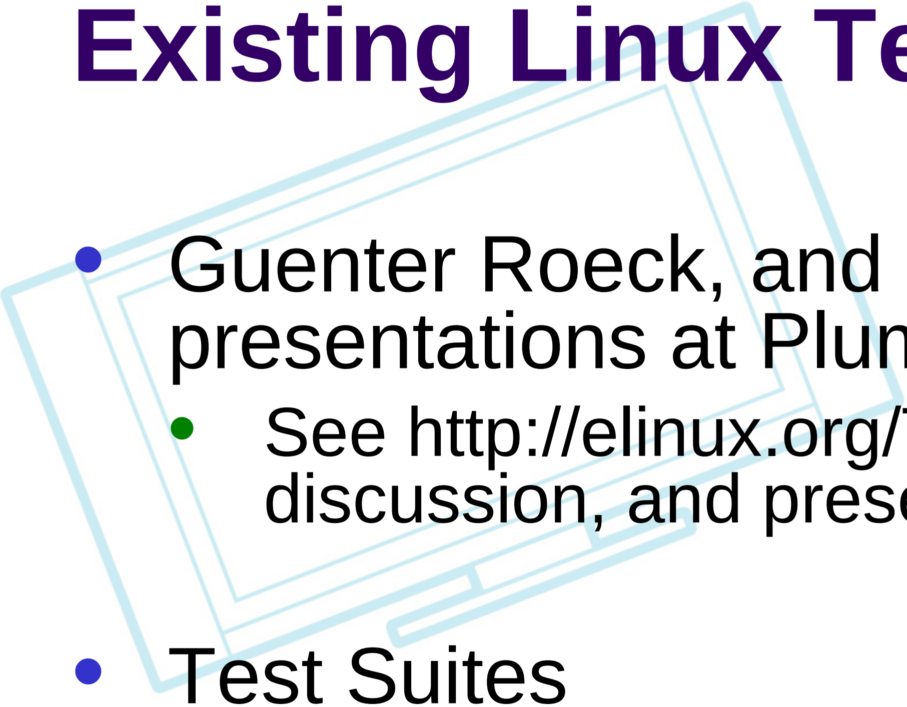
Existing Linux Tests

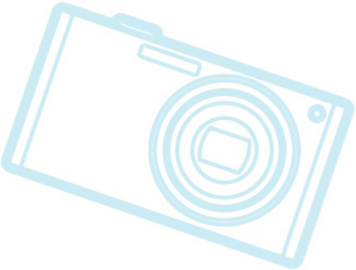
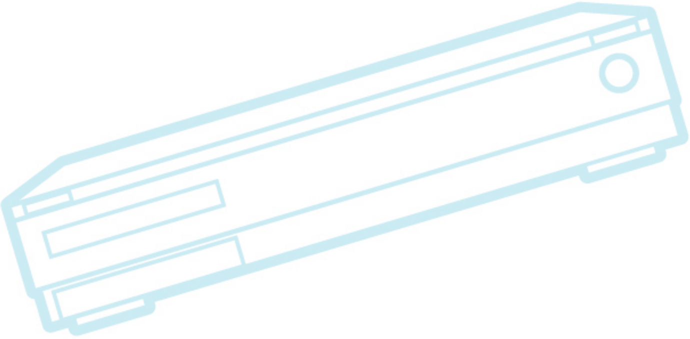




Existing Linux Tests

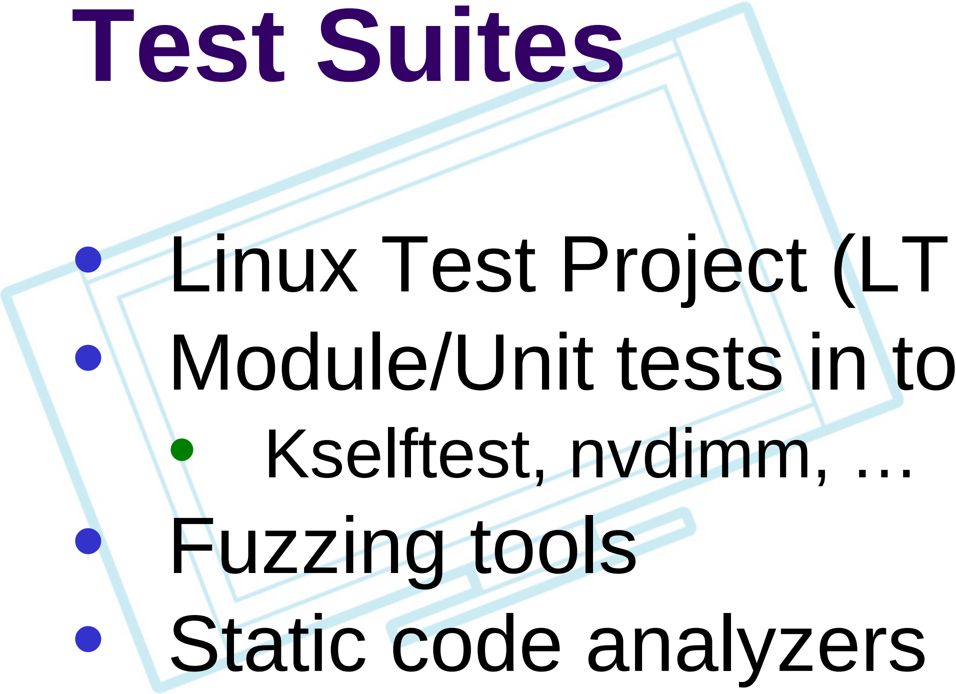
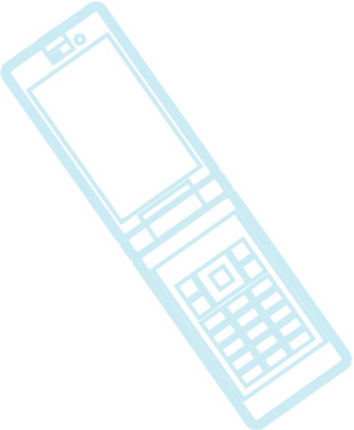
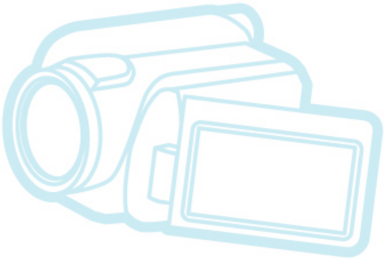
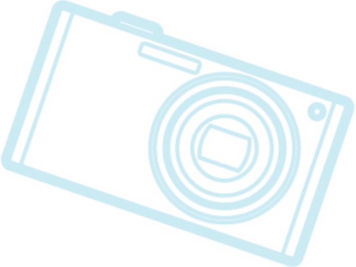
- Guenter Roeck, and others, did presentations at Plumbers Conference 2016
  - See http://elinux.org/Testing for etherpad of discussion, and presentation links
- Test Suites
- Test Beds



Test Suites

- Linux Test Project (LTP)
- Module/Unit tests in tools/testing
  - Kselftest, nvdimm, …
- Fuzzing tools
- Static code analyzers
- Individual tests
  - Filesystem, networking, vulnerability, driver, etc.

LTP

- Big collection of tests
  - 1000+ system calls
  - 1000+ POSIX conformance tests
  - 400+ IO stress tests
  - Realtime, networking, cgroups, namespace tests
- Has been around a long time
- https://github.com/linux-test-project/ltp/wiki
- Common to find failures you don’t care about
  - What is your experience?

Kernel Self Test

- Unit test framework in the Linux kernel
- Integrated with kernel source
  - /tools/testing/selftests
- See https://lwn.net/Articles/608959/
- No unified output
- Very ad-hoc organization

Fuzzing Tools

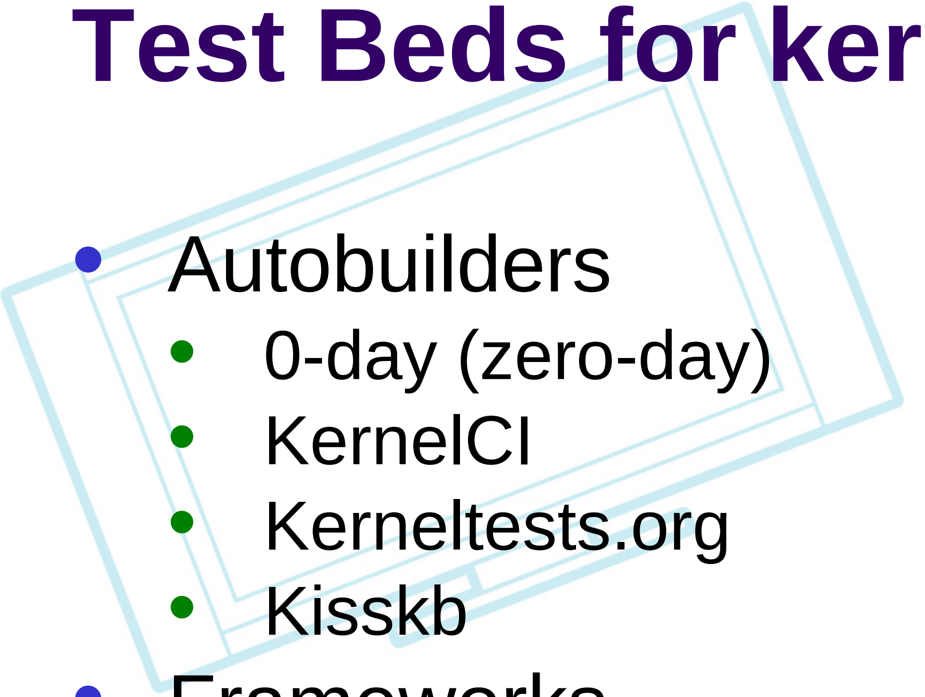
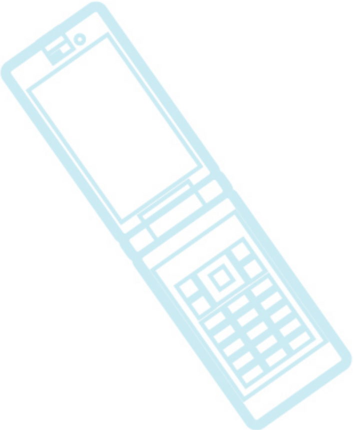
- Trinity
  - Linux System Call fuzz tester
  - http://codemnkey.org.uk/projects/trinity/
- Syzkaller
  - Coverage-guided Linux syscall fuzzer
  - Meant to be used with KASAN
  - https://github.com/google.syzkaller

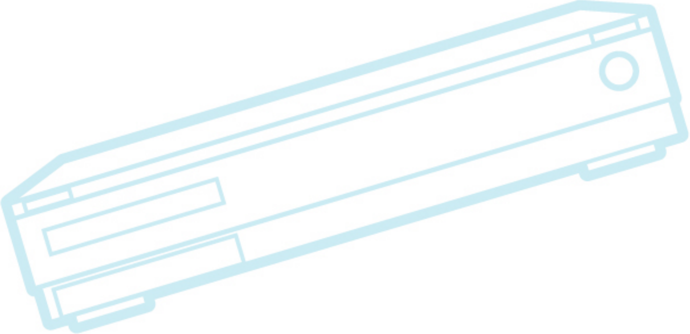
Static Code analyzers

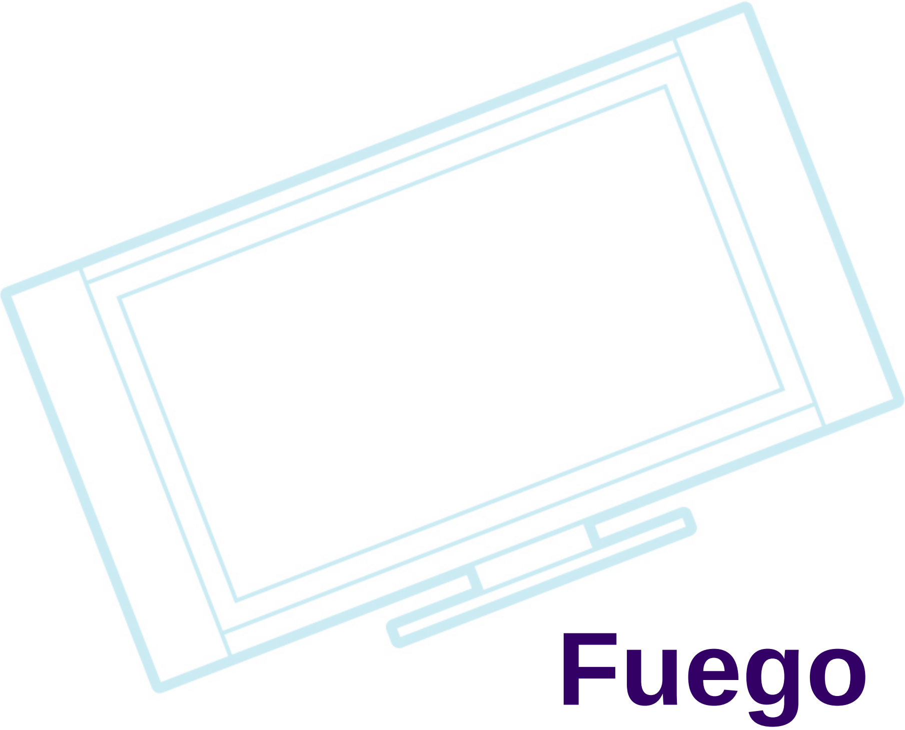
- Coccinelle
  - Program matching and transformation engine
- Coverity
  - Commercial static analyzer
  - Coverity company runs the tool against the Linux kernel continually
- gcc warnings
- Smatch
- Sparse

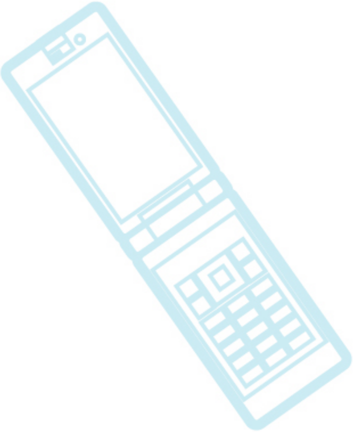
Test Beds for kernel

- Autobuilders
  - 0-day (zero-day)
  - KernelCI
  - Kerneltests.org
  - Kisskb
- Frameworks
  - Jenkins
  - Lava
  - Avacado
  - Fuego
  - Buildbot

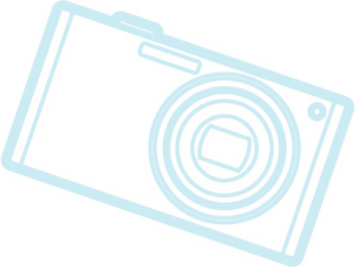



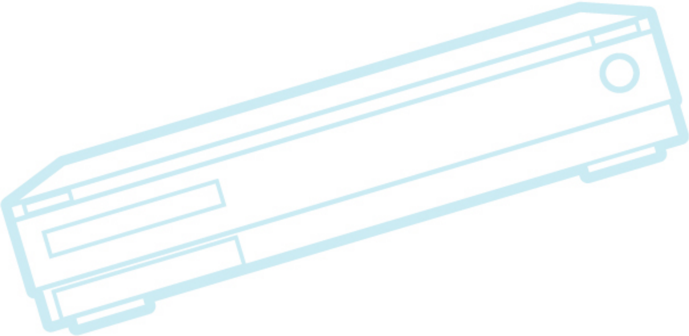






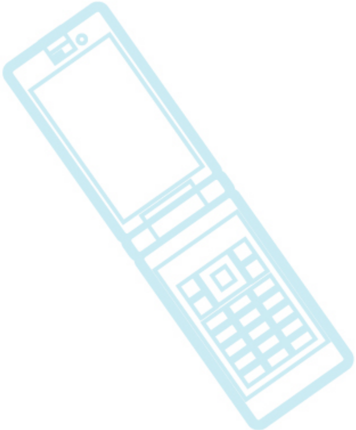
Fuego Roadmap Update



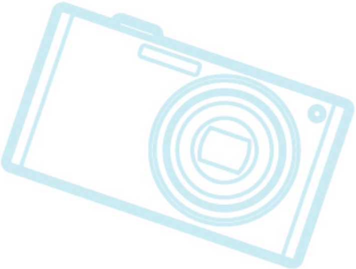


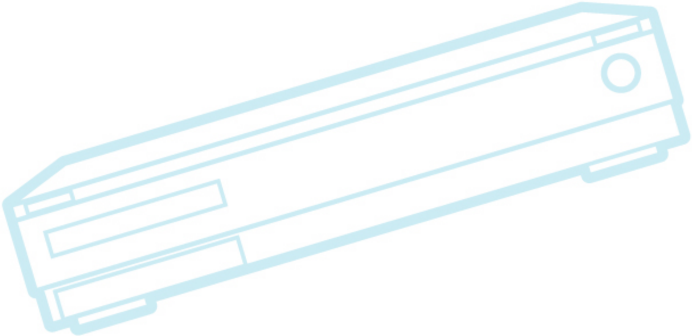
Fuego Roadmap Update

- 1.2 Release
  - Major features
    - Test Dependencies
    - LAVA support
    - Unified output format
    - Documentation update
  - New tests:
    - Kselftest
    - Fuego unit tests
    - AGL tests
  - Planned for summer 2017 (mid-July rc1??)
- After that:
  - Test hub – test store, test results database, public distributed test queue
  - Much more…



Discussion Points





Issues to discuss

- Ability to share tests
  - How to build, how to run
- Ability to share results
  - Standard format, interpretation of results, communicating with upstream
- Ability to share and collaborate on sub-tools
  - Board reboot, power control, provisioning, monitoring, power reporting
- Standards
  - Output format standard
    - Fuego and KernelCI have communicated
- Creating a collaboration economy for QA knowledge and assets

Issues during Q&A

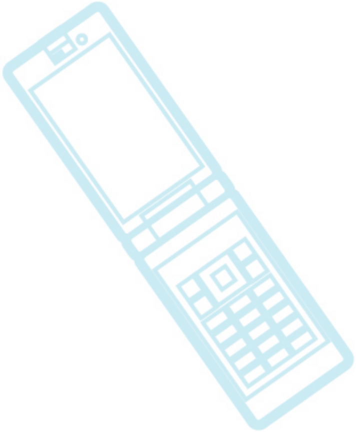
- Big obstacle to testing is knowing how to interpret results. Does Fuego address this?
  - Unified output makes sharing results easier
  - Shared results has potential to allow identification of “real” errors, vs. things that are failing for other reasons (expected failures, config, board setup, etc.)
- What about security testing?
  - It’s important, OpenVAS and VULS were discussed at OSSJ.
  - OpenVAS requires distribution
    - May need changes to support custom distro like one built with Yocto

Issues during Q&A (cont.)

- What about complex pattern matching – any solution?
  - CVE testing needs to scan a lot of text, and compare with a lot of patterns
  - Fuego pattern matching is simple – does not currently address this issue
- Different projects would benefit from output standard
  - This is something distro testers want because of large number of packages with different test output formats
  - Been discussing for a long time, but no one starts it
  - Maybe can establish de-facto standard with just a few key projects (e.g. KernelCI, Avacado, Fuego)

Issues during Q&A (cont. 2)

- Lightning talk on ktest – a tool to automate testing tasks for the Linux kernel
  - Already in Linux kernel tree
  - Gives ability to execute tests that same no matter what the target
  - Supports commit bisect, config bisect, commit reverse bisect



Thanks