

A background graphic consisting of a network of blue lines connecting various yellow circular nodes, set against a dark blue gradient background.

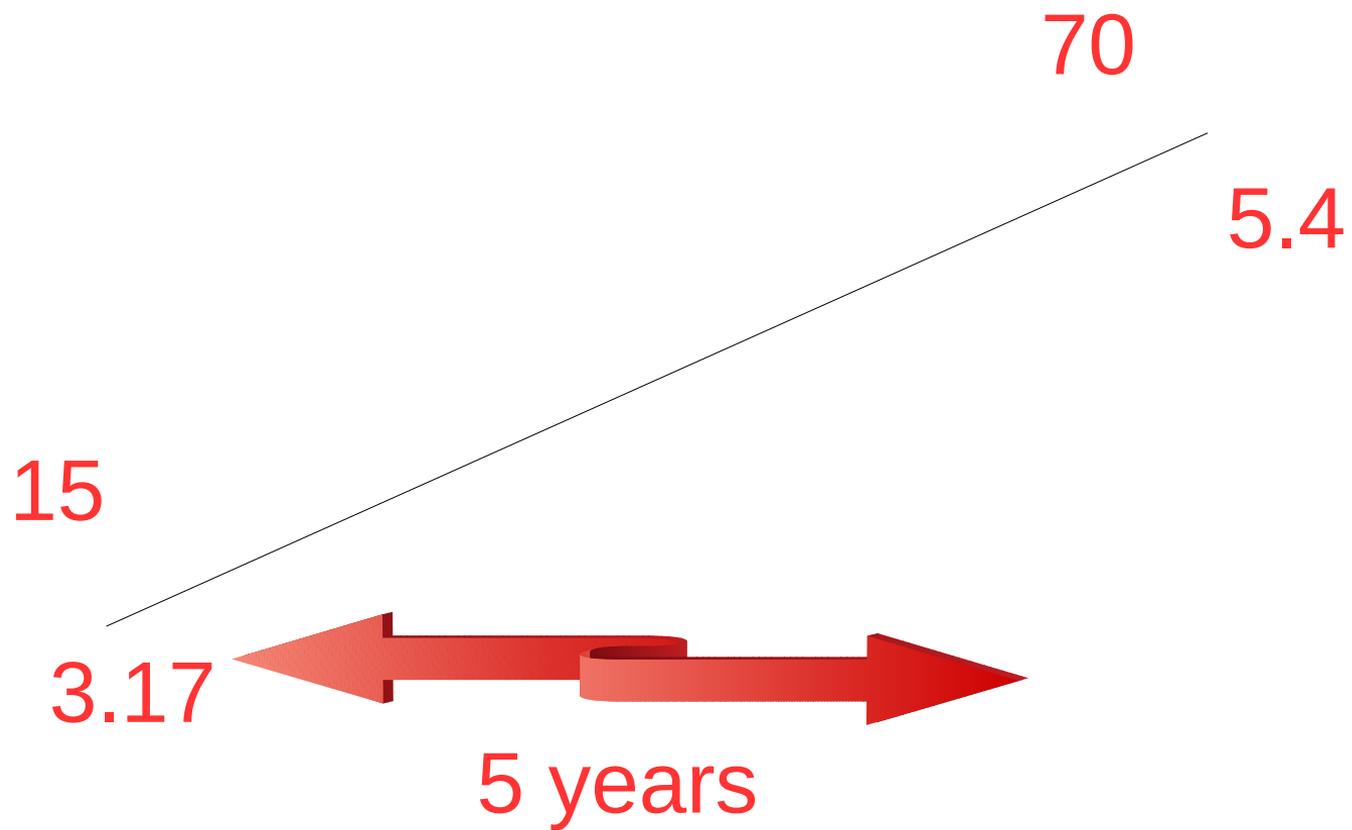
Kselftest LPC and Kernel Summit 2019

Shuah Khan

Linux Kernel Fellow

skhan@linuxfoundation.org

 THE **LINUX** FOUNDATION



Linux Kernel Functional Testing (LKFT)

- Architectures: arm32, arm64, i386, x86_64
- Hardware: X15, DragonBoard 410c, Juno, HiKey, x86_64 servers,
- QEMU: x86* on x86_64 servers, arm* on SynQuacer arm64 hosts
- Linux Branches:
 - LTS: 4.4, 4.9, 4.14, 4.19
 - Latest stable (5.2), mainline, next
- Tests: LTP, libhugetlbfs, perf, v4l2, kvm-unit-tests, s-suite (i/o benchmark), kselftests
- All tests run in all environments on every push for a total of ~25,000 tests per push.
- **Authors - Dan Rue and Anders Roxell**



Kselftest @



Selected results ↴

Linux Version	Kselftest Version
next	next
mainline	mainline
4.4.x, 4.9.x, 4.14.x, 4.19.x	v5.2.11 (latest stable release)

Board	v5.3-rc7, in-kernel selftest			v4.9.190, v5.1 selftest			v4.9.190, in-kernel selftest	
	Pass	Skip	Fail	Pass	Skip	Fail	Pass	Fail
qemu_x86_64	126	17	55	85	28	66	45	8
x86_64	129	13	57	82	26	63	44	8
qemu_i386	115	16	65	89	17	78	50	9
i386	84	13	57	84	16	71	49	8
qemu_arm64	102	15	63	66	25	74	34	6
juno-r2 - arm64	99	14	63	66	24	74	29	6
db410c - arm64	100	15	65	66	24	71	32	7
hikey - arm64	100	15	63	66	21	74	28	6
qemu_arm	94	14	70	66	24	74	33	6
x15 - arm	97	15	68	68	23	83	34	6
Total	1046	147	626	738	228	728	378	70

Kselftest use-cases ...

Kernel and Kselftest rev matched

Linux kernel

Kselftest from latest stable

Stable release

Install and run Kselftest on a target

Discussion agenda

- Kselftest use-case discussion.
 - Run latest stable kselftest on stable kernels
 - Proving to be problematic. Balance coverage vs. dealing with test bugs in failing to skip when they should.
 - See large number of skips – this doesn't necessarily offer coverage. Hard to measure how much more coverage we get from this option.
 - Some areas such as networking and bpf are very active and often the reason for increased test failures.
 - All but bpf support latest stable kselftest on stable kernels. This is an exception and not a rule.

Results from stable release

- 5.3.rc8
 - qemu_x86_64 (198 total: 125p, 17skip, 21fail, 35 known fails) – 63% 8.5%skip **11%fail, 18%known fails**
 - qemu_arm64 (180 total: 102p, 15skip, 30fail, 33known fails) - 57%p, 8%skip, **17%, 18%known fails**

Results from stable release

- 5.2.14-rc1
 - qemu_x86_64 (178 total: 129p, 12skip, 7fail, 30known fails) - 72%p, 6.7%skip, **3.9% fail, 16%known fails**
 - qemu_arm64 (168 total: 111p, 17skip, 8fail, 32known fails) - 66%p, 10%skip, **4.7% fail, 19%known fails**
- 4.19.72-rc1
 - qemu_x86_64 (181 total: 117p, 16skip, 2fail, 46known fails) – 65%p, 8%skip, **1%fail, 25%known fails**
 - qemu_arm64 (168 total: 94p, 24skip, 1fail, 49known fails) – 55%p, 14%skip, **0%fail, 29%known fails**

Results from stable release

- 4.14.143-rc1
 - qemu_x86_64 (178 total: 90p, 19skip, 2fail, 67known fails) – 50%p, 10%skip, **1%fail, 37%known fails**
 - qemu_arm64 (166 total:72p, 24skip, 1fail, 69known fails) – 43%p, 14%skip, **0.6%fail, 41%known fails**
- 4.9.192-rc1
 - qemu_x86_64 (179 total: 85p, 29skip, 3fail, 62known fails) – 47%p, 16%skip, **.01%fail, 35%known fails**
 - qemu_arm64 (165 total: 66p, 27skip, 1fail, 71known fails) – 40%p, 16%skip, **0%fail, 43%known fails**

Next steps ...

- Improve process – find test bugs early before they make it into a release. linux-next reports during merge window.
- Need help and support from individual test maintainers to reduce overhead on CI ring admins in dealing with breakages.
- Reduce known fails and fails numbers.

Next steps ...

- Since my talk calling out ARM coverage as weak, several ARM tests are added. Still more work is needed.
- Continue to increase coverage. Driver coverage is weak.
- Improve framework. Clearly identify dependencies on other utilities and recommend install options. e.g x86 test 32-bit library dependency checks.
- Kbuild integration? – easier if kselftest gets moved to root level. Being under tools is making it difficult to add support. (patch flow to stable could become harder with the move) – Discussed and developers don't want this feature.