Checking your work: Linux kernel testing and CI

David Vernet
void@manifault.com

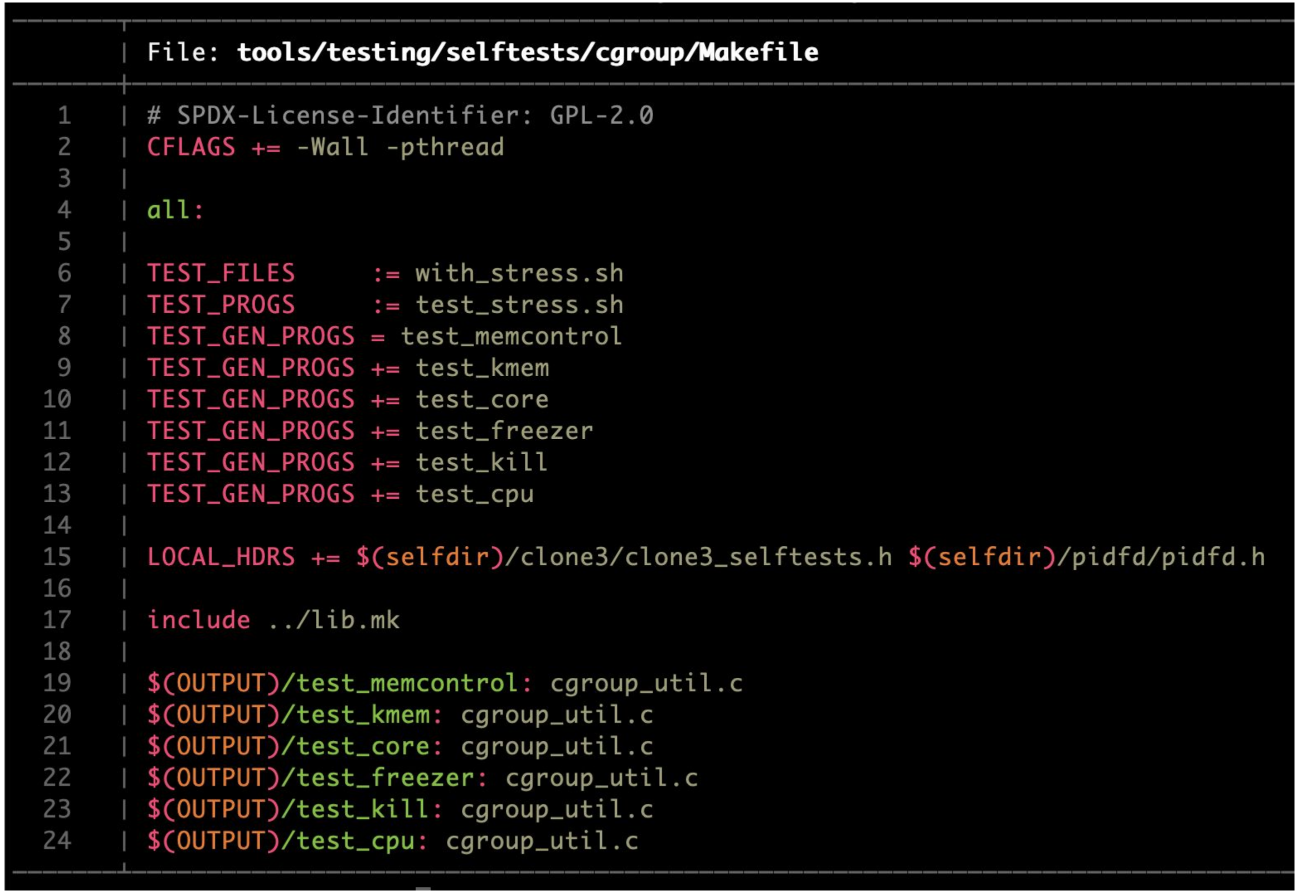
Linux Plumbers Conference 2022 – Dublin, Ireland

Agenda

What are kselftests?

- A flexible testing framework for validating the Linux kernel
- Testcases are instances of userspace programs
    - Commonly written in C, but need only be an executable file
    - Often output results in KTAP format
- Located in tree at `tools/testing/selftests`
- Many different subsystems tested, all (seemingly) slightly differently

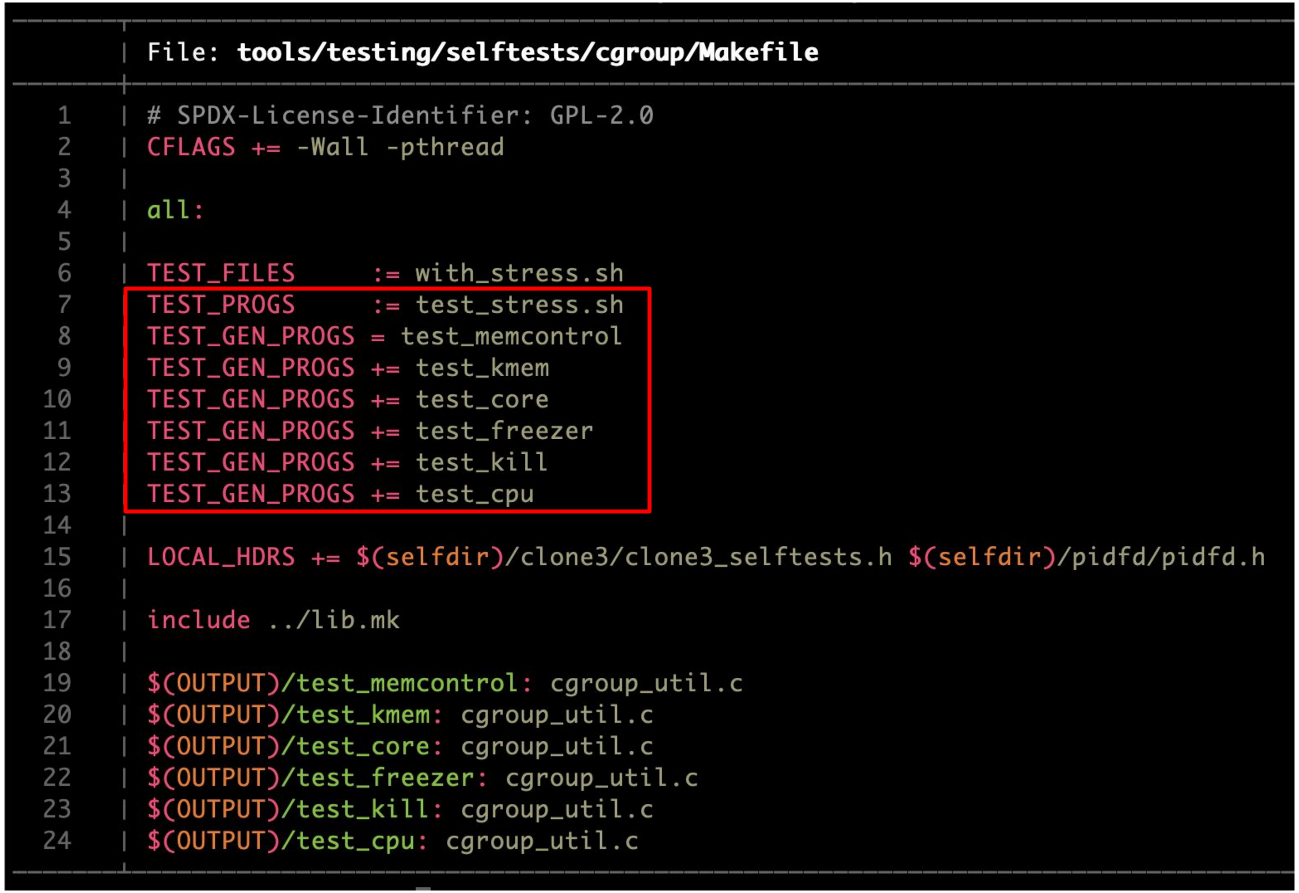
A kselftest suite is defined by its Makefile



```
File: tools/testing/selftests/cgroup/Makefile
```

```
# SPDX-License-Identifier: GPL-2.0
CFLAGS += -Wall -pthread

all:

TEST_FILES     := with_stress.sh
TEST_PROGS     := test_stress.sh
TEST_GEN_PROGS = test_memcontrol
TEST_GEN_PROGS += test_kmem
TEST_GEN_PROGS += test_core
TEST_GEN_PROGS += test_freezer
TEST_GEN_PROGS += test_kill
TEST_GEN_PROGS += test_cpu

LOCAL_HDRS += $(selfdir)/clone3/clone3_selftests.h $(selfdir)/pidfd/pidfd.h

include ../lib.mk

$(OUTPUT)/test_memcontrol: cgroup_util.c
$(OUTPUT)/test_kmem: cgroup_util.c
$(OUTPUT)/test_core: cgroup_util.c
$(OUTPUT)/test_freezer: cgroup_util.c
$(OUTPUT)/test_kill: cgroup_util.c
$(OUTPUT)/test_cpu: cgroup_util.c
```

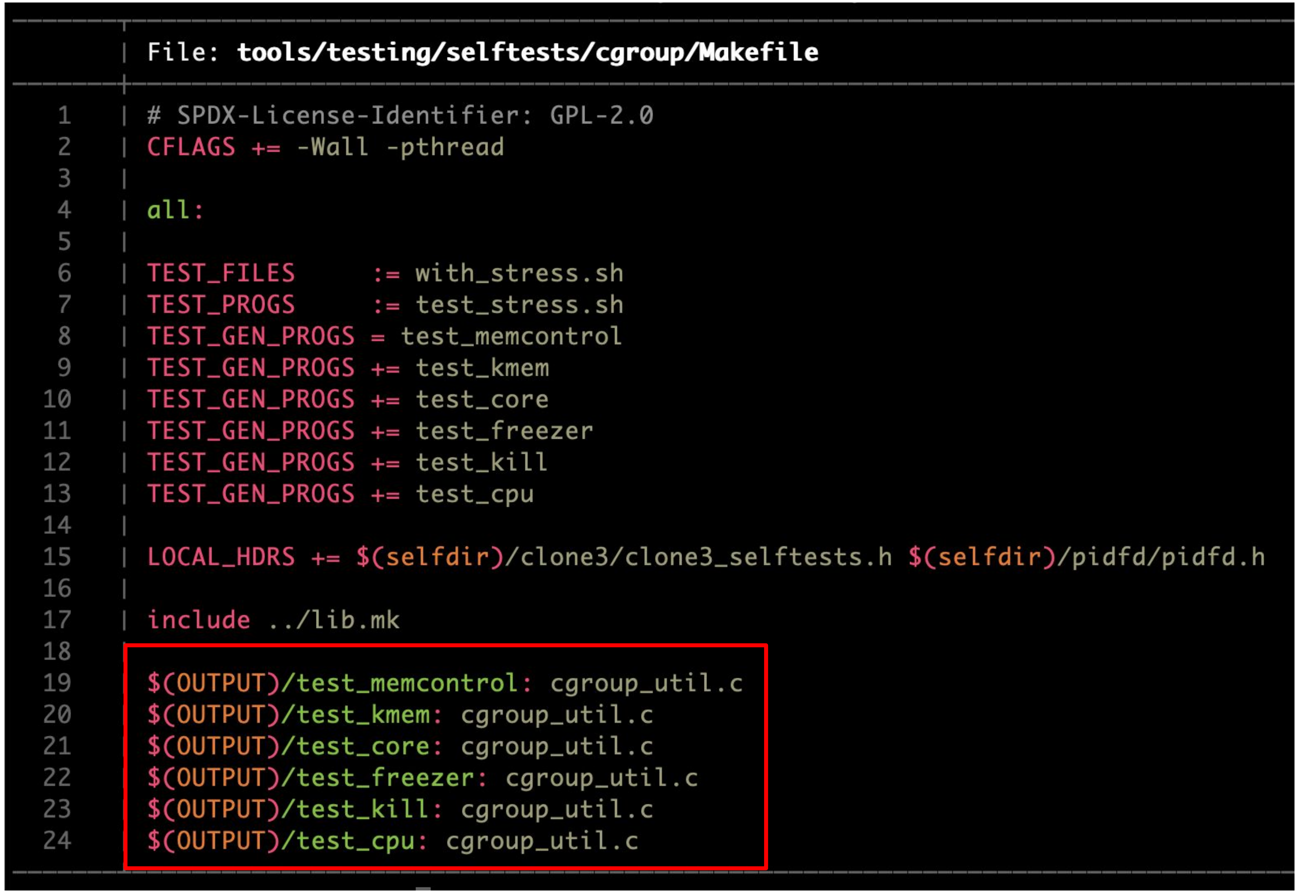
A kselftest suite is defined by its Makefile



File: **tools/testing/selftests/cgroup/Makefile**

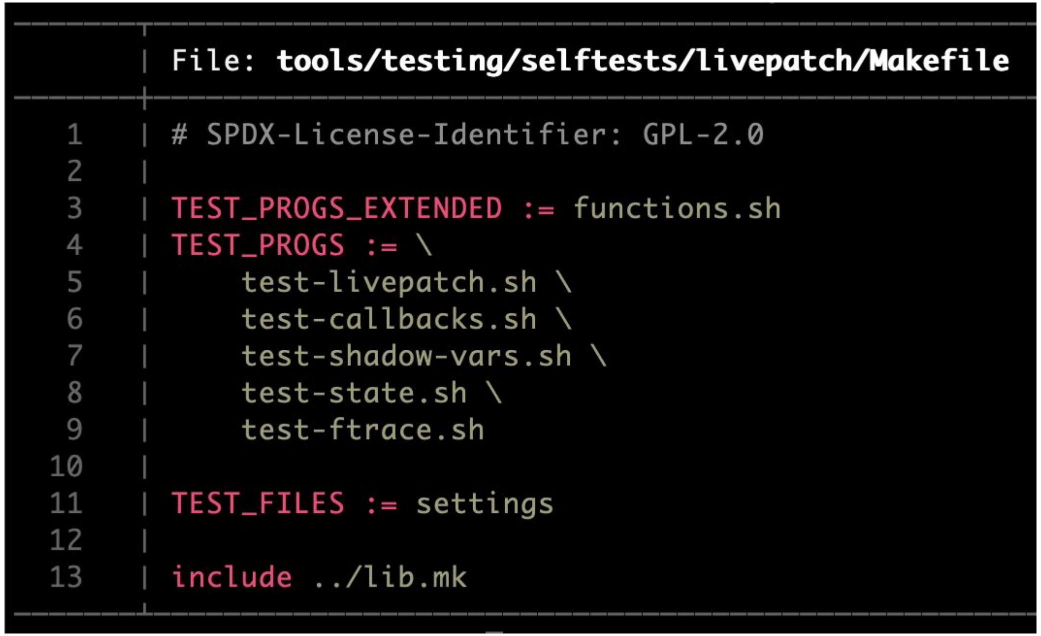
```
# SPDX-License-Identifier: GPL-2.0
CFLAGS += -Wall -pthread

all:

TEST_FILES     := with_stress.sh
TEST_PROGS     := test_stress.sh
TEST_GEN_PROGS = test_memcontrol
TEST_GEN_PROGS += test_kmem
TEST_GEN_PROGS += test_core
TEST_GEN_PROGS += test_freezer
TEST_GEN_PROGS += test_kill
TEST_GEN_PROGS += test_cpu

LOCAL_HDRS += $(selfdir)/clone3/clone3_selftests.h $(selfdir)/pidfd/pidfd.h

include ../lib.mk

$(OUTPUT)/test_memcontrol: cgroup_util.c
$(OUTPUT)/test_kmem: cgroup_util.c
$(OUTPUT)/test_core: cgroup_util.c
$(OUTPUT)/test_freezer: cgroup_util.c
$(OUTPUT)/test_kill: cgroup_util.c
$(OUTPUT)/test_cpu: cgroup_util.c
```

A kselftest suite is defined by its Makefile

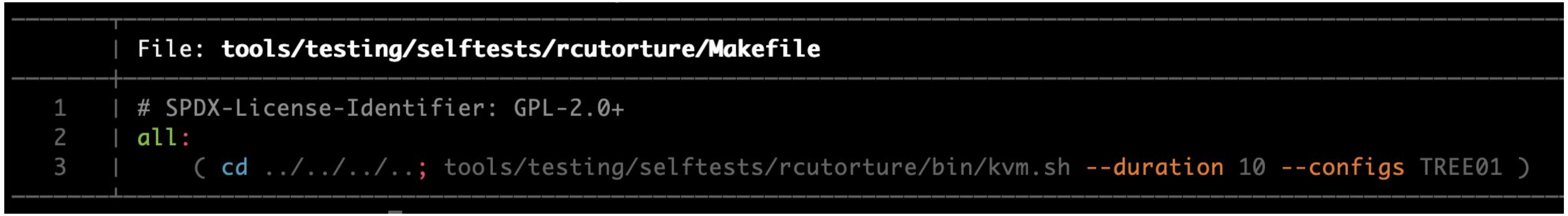


```
File: tools/testing/selftests/cgroup/Makefile

# SPDX-License-Identifier: GPL-2.0
CFLAGS += -Wall -pthread

all:

TEST_FILES     := with_stress.sh
TEST_PROGS     := test_stress.sh
TEST_GEN_PROGS = test_memcontrol
TEST_GEN_PROGS += test_kmem
TEST_GEN_PROGS += test_core
TEST_GEN_PROGS += test_freezer
TEST_GEN_PROGS += test_kill
TEST_GEN_PROGS += test_cpu

LOCAL_HDRS += $(selfdir)/clone3/clone3_selftests.h $(selfdir)/pidfd/pidfd.h

include ../lib.mk

$(OUTPUT)/test_memcontrol: cgroup_util.c
$(OUTPUT)/test_kmem: cgroup_util.c
$(OUTPUT)/test_core: cgroup_util.c
$(OUTPUT)/test_freezer: cgroup_util.c
$(OUTPUT)/test_kill: cgroup_util.c
$(OUTPUT)/test_cpu: cgroup_util.c
```

A kselftest suite is defined by its Makefile



```
File: tools/testing/selftests/livepatch/Makefile

# SPDX-License-Identifier: GPL-2.0

TEST_PROGS_EXTENDED := functions.sh
TEST_PROGS := \
	test-livepatch.sh \
	test-callbacks.sh \
	test-shadow-vars.sh \
	test-state.sh \
	test-ftrace.sh

TEST_FILES := settings

include ../lib.mk
```



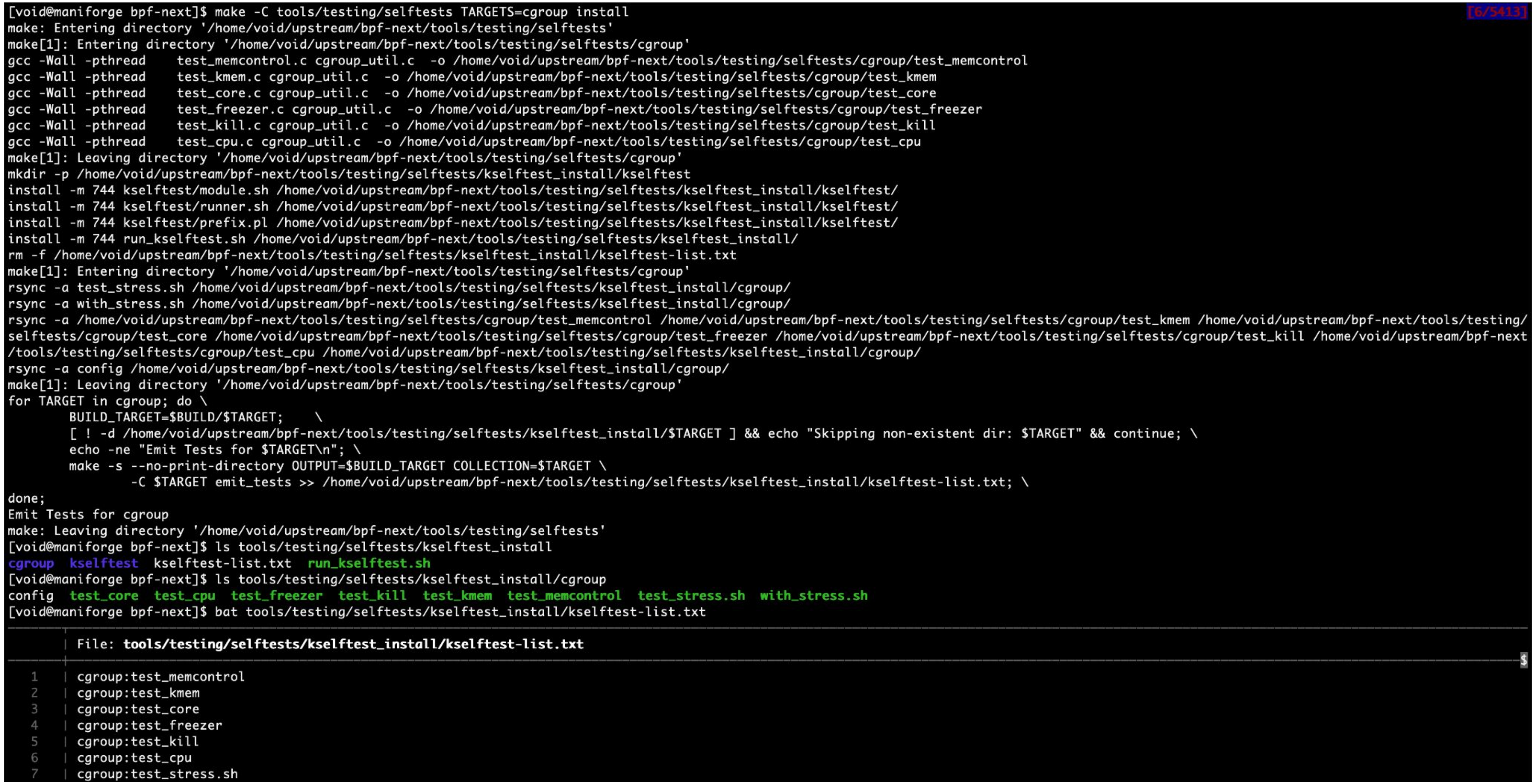
```
File: tools/testing/selftests/rcutorture/Makefile

# SPDX-License-Identifier: GPL-2.0+
all:
	( cd ../../../..; tools/testing/selftests/rcutorture/bin/kvm.sh --duration 10 --configs TREE01 )
```

kselftests framework is intentionally very flexible

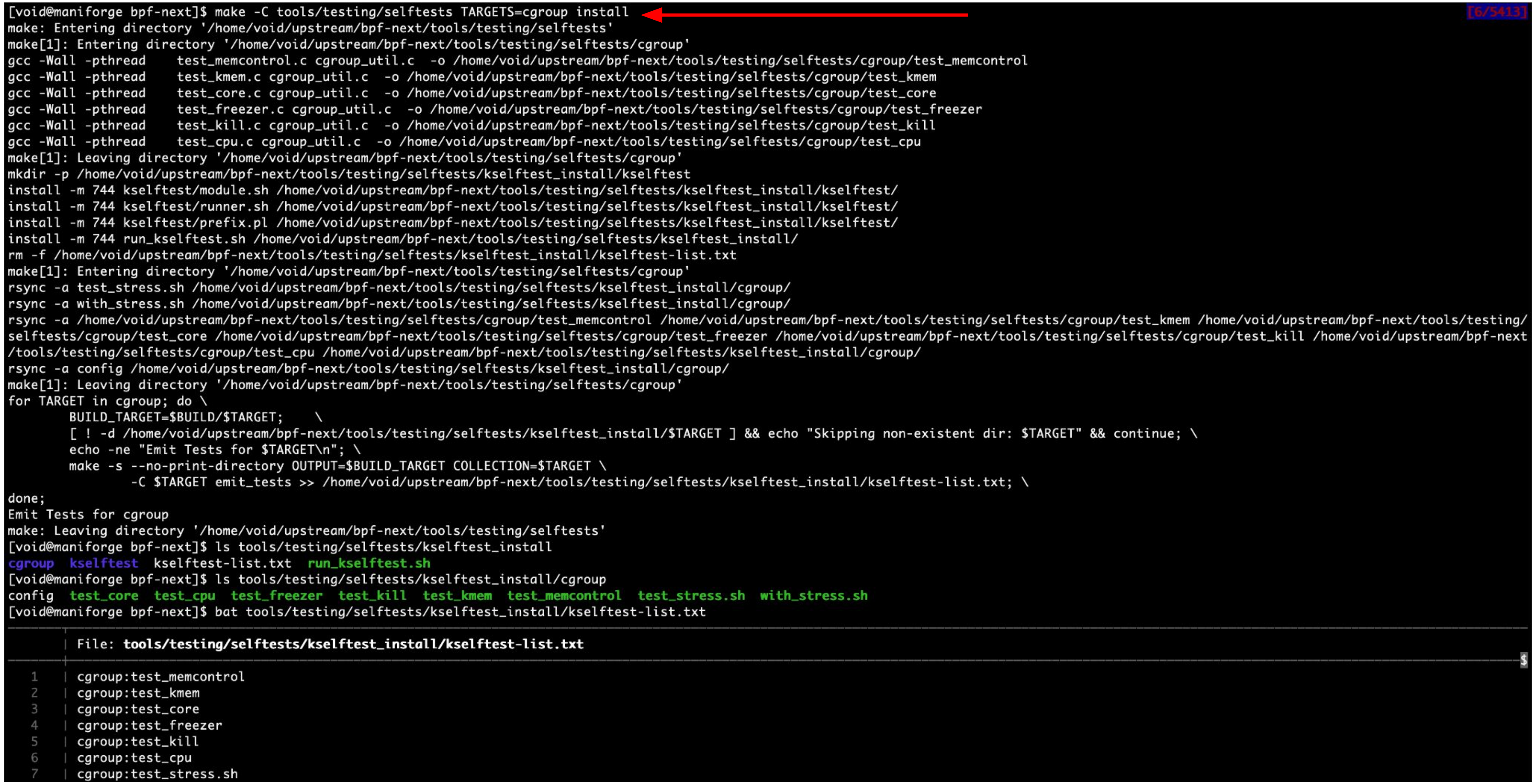
- kselftests are only **required** to define a Makefile
- Otherwise, the suite can do anything
  - E.g. define a single target which runs a shell script that loads a module that does heavy lifting (RCU)
  - E.g. specify a few executable targets that function as testcases (livepatch)
  - E.g. specify some targets that are compiled from .c files, and then run as testcases (cgroup)
  - E.g. a combination of specifying testcases (TEST_GEN_PROGS, TEST_PROGS), and a single shell script that is responsible for invoking testcases.

kselftests can be built, installed, and run

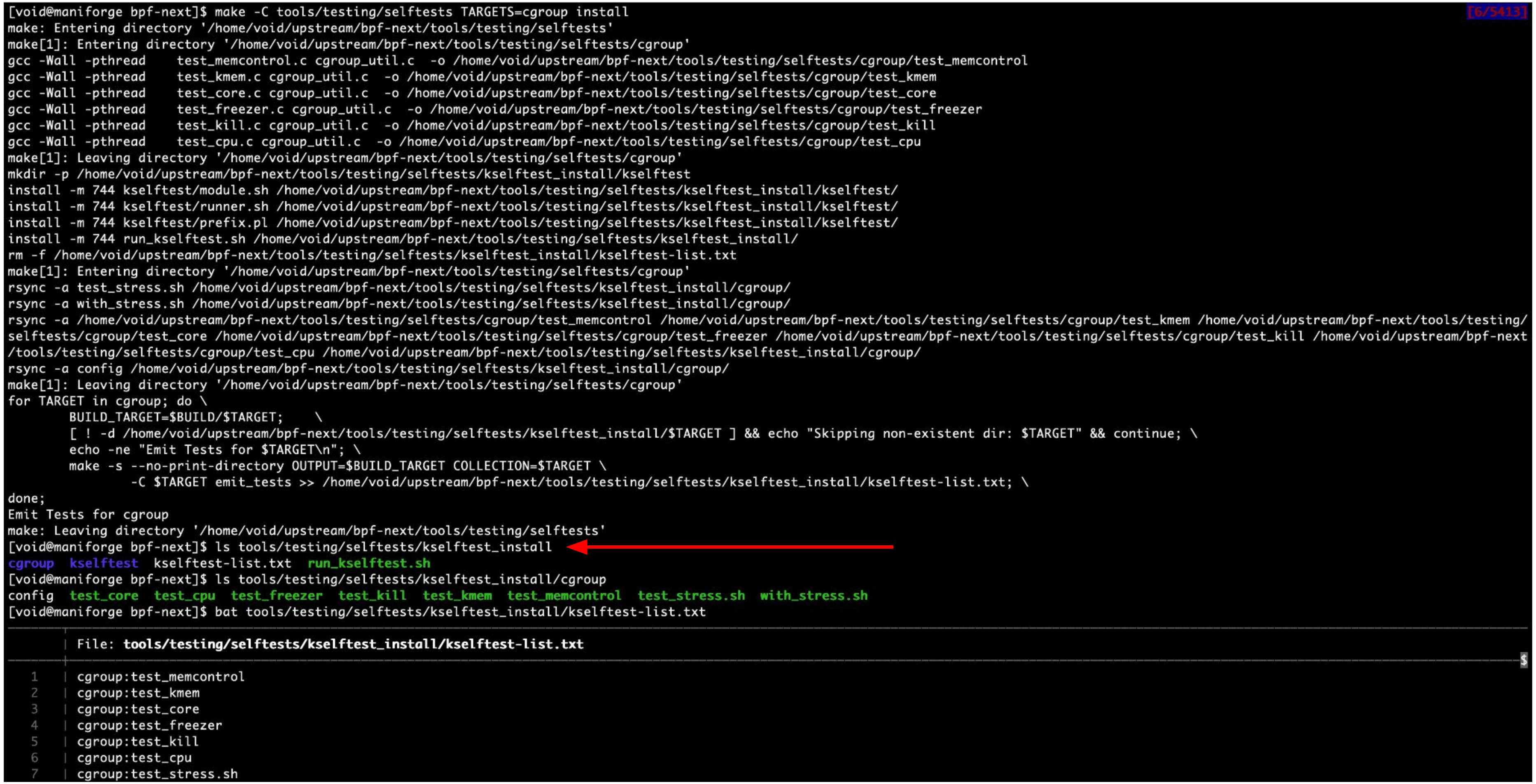
- Details about this can be found on the kselftest kernel doc page
  - https://docs.kernel.org/dev-tools/kselftest.html
  - https://kselftest.wiki.kernel.org/
- *Installing* builds one or more specified test-suites, packages the output executables, and creates a “test runner” that can invoke the tests on your behalf
  - `make -C tools/testing/selftests TARGETS=”...” install`
  - Builds all test targets, and outputs them into a kselftest_install directory
- Creates a test runner that can be invoked to run all of the tests



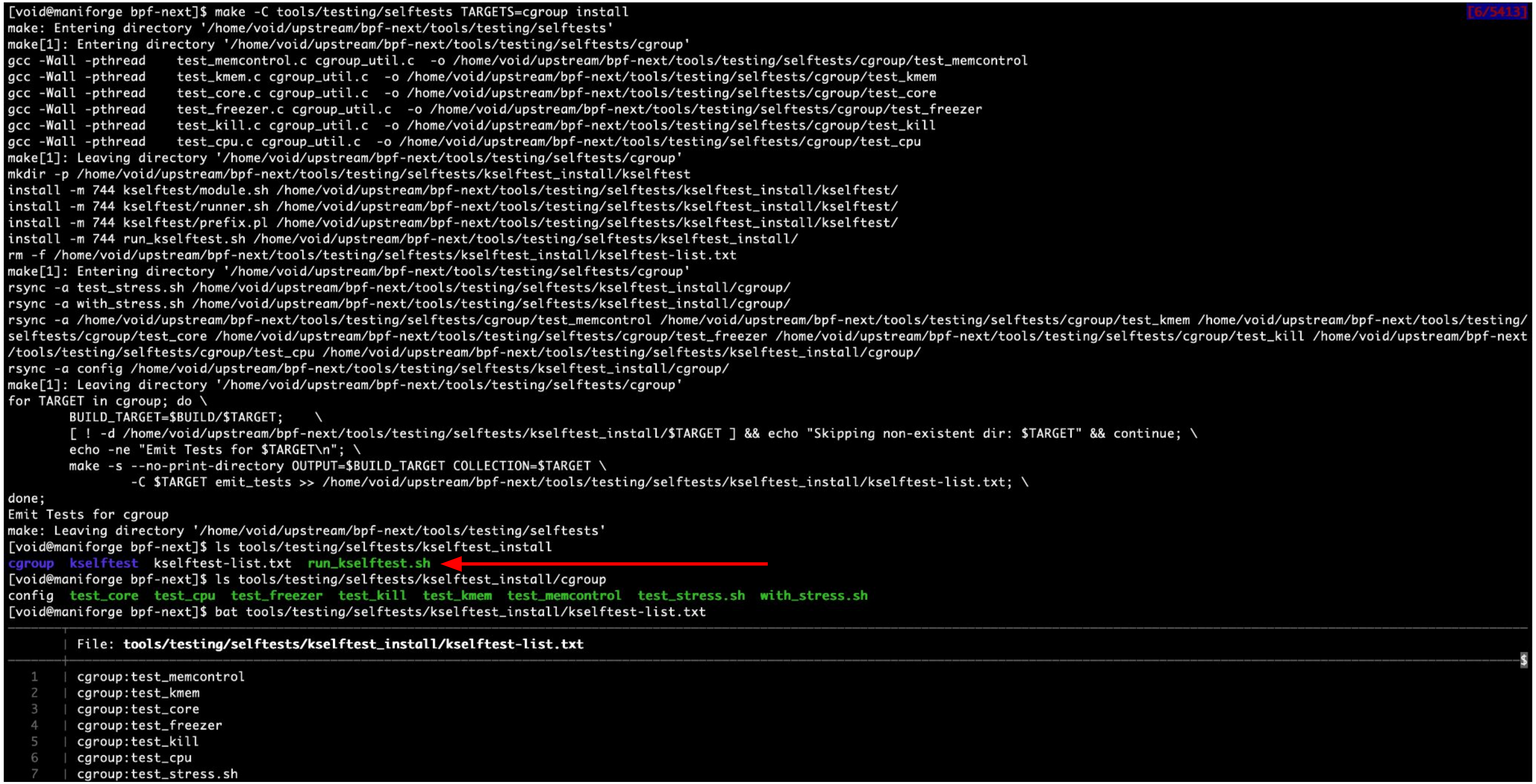
```
[void@maniforge bpf-next]$ make -C tools/testing/selftests TARGETS=cgroup install                                    [6/5413]
make: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests'
make[1]: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
gcc -Wall -pthread    test_memcontrol.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_memcontrol
gcc -Wall -pthread    test_kmem.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kmem
gcc -Wall -pthread    test_core.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_core
gcc -Wall -pthread    test_freezer.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_freezer
gcc -Wall -pthread    test_kill.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kill
gcc -Wall -pthread    test_cpu.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_cpu
make[1]: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
mkdir -p /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest
install -m 744 kselftest/module.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 kselftest/runner.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 kselftest/prefix.pl /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 run_kselftest.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/
rm -f /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest-list.txt
make[1]: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
rsync -a test_stress.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a with_stress.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_memcontrol /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kmem /home/void/upstream/bpf-next/tools/testing/
selftests/cgroup/test_core /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_freezer /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kill /home/void/upstream/bpf-next
/tools/testing/selftests/cgroup/test_cpu /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a config /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
make[1]: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
for TARGET in cgroup; do \
        BUILD_TARGET=$BUILD/$TARGET;    \
        [ ! -d /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/$TARGET ] && echo "Skipping non-existent dir: $TARGET" && continue; \
        echo -ne "Emit Tests for $TARGET\n"; \
        make -s --no-print-directory OUTPUT=$BUILD_TARGET COLLECTION=$TARGET \
                -C $TARGET emit_tests >> /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest-list.txt; \
done;
Emit Tests for cgroup
make: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests'
[void@maniforge bpf-next]$ ls tools/testing/selftests/kselftest_install
cgroup  kselftest  kselftest-list.txt  run_kselftest.sh
[void@maniforge bpf-next]$ ls tools/testing/selftests/kselftest_install/cgroup
config  test_core  test_cpu  test_freezer  test_kill  test_kmem  test_memcontrol  test_stress.sh  with_stress.sh
[void@maniforge bpf-next]$ bat tools/testing/selftests/kselftest_install/kselftest-list.txt
───────┬────────────────────────────────────────────────────────────────────────────────
       │ File: tools/testing/selftests/kselftest_install/kselftest-list.txt
───────┼────────────────────────────────────────────────────────────────────────────────
       │ cgroup:test_memcontrol
       │ cgroup:test_kmem
       │ cgroup:test_core
       │ cgroup:test_freezer
       │ cgroup:test_kill
       │ cgroup:test_cpu
       │ cgroup:test_stress.sh
```



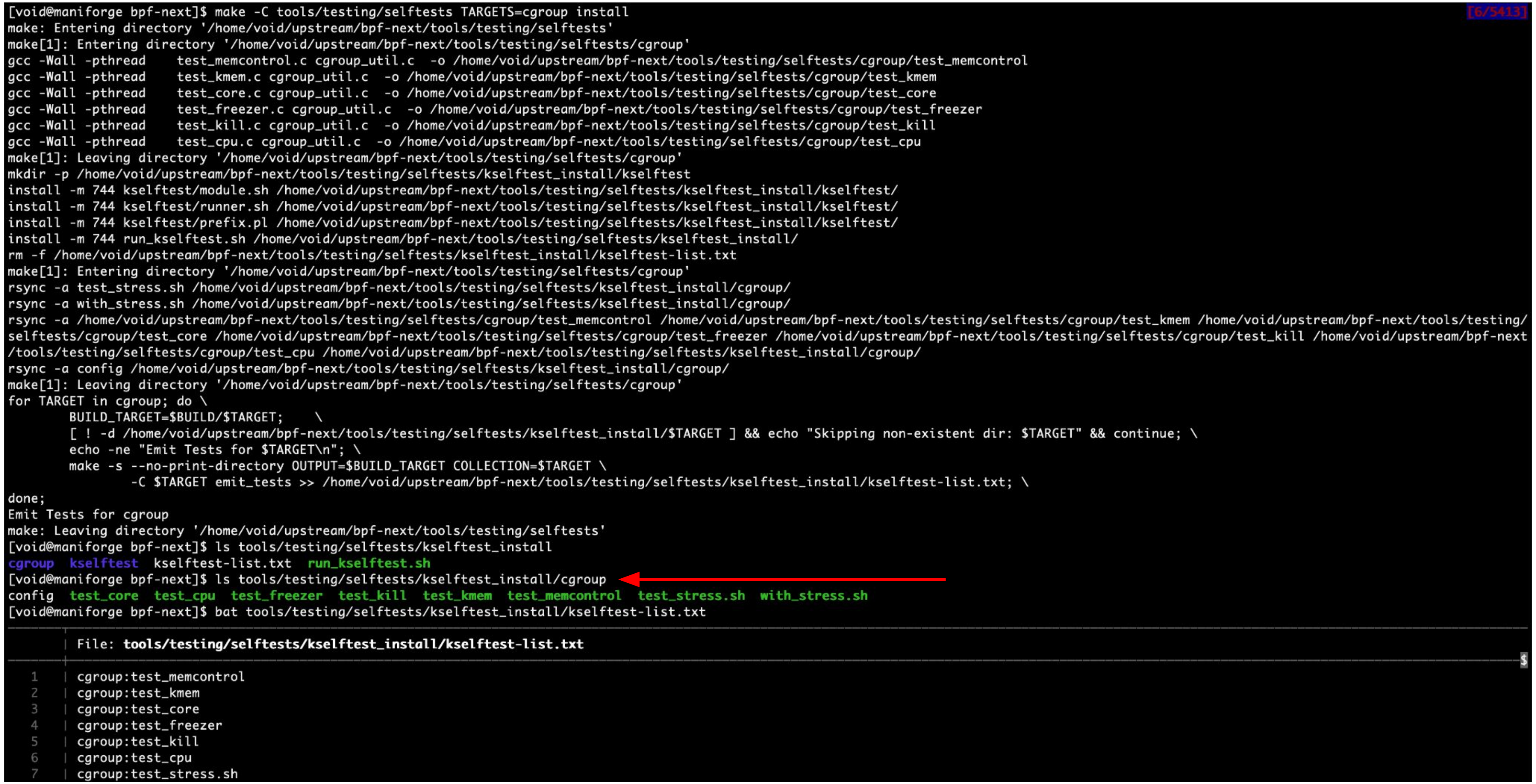
```
[void@maniforge bpf-next]$ make -C tools/testing/selftests TARGETS=cgroup install                                                      [6/5413]
make: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests'
make[1]: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
gcc -Wall -pthread    test_memcontrol.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_memcontrol
gcc -Wall -pthread    test_kmem.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kmem
gcc -Wall -pthread    test_core.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_core
gcc -Wall -pthread    test_freezer.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_freezer
gcc -Wall -pthread    test_kill.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kill
gcc -Wall -pthread    test_cpu.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_cpu
make[1]: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
mkdir -p /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest
install -m 744 kselftest/module.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 kselftest/runner.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 kselftest/prefix.pl /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 run_kselftest.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/
rm -f /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest-list.txt
make[1]: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
rsync -a test_stress.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a with_stress.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_memcontrol /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kmem /home/void/upstream/bpf-next/tools/testing/
selftests/cgroup/test_core /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_freezer /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kill /home/void/upstream/bpf-next
/tools/testing/selftests/cgroup/test_cpu /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a config /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
make[1]: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
for TARGET in cgroup; do \
        BUILD_TARGET=$BUILD/$TARGET;    \
        [ ! -d /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/$TARGET ] && echo "Skipping non-existent dir: $TARGET" && continue; \
        echo -ne "Emit Tests for $TARGET\n"; \
        make -s --no-print-directory OUTPUT=$BUILD_TARGET COLLECTION=$TARGET \
                -C $TARGET emit_tests >> /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest-list.txt; \
done;
Emit Tests for cgroup
make: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests'
[void@maniforge bpf-next]$ ls tools/testing/selftests/kselftest_install
cgroup  kselftest  kselftest-list.txt  run_kselftest.sh
[void@maniforge bpf-next]$ ls tools/testing/selftests/kselftest_install/cgroup
config  test_core  test_cpu  test_freezer  test_kill  test_kmem  test_memcontrol  test_stress.sh  with_stress.sh
[void@maniforge bpf-next]$ bat tools/testing/selftests/kselftest_install/kselftest-list.txt
───────┬────────────────────────────────────────────────────────────────────────
       │ File: tools/testing/selftests/kselftest_install/kselftest-list.txt
───────┼────────────────────────────────────────────────────────────────────────
       │ cgroup:test_memcontrol
       │ cgroup:test_kmem
       │ cgroup:test_core
       │ cgroup:test_freezer
       │ cgroup:test_kill
       │ cgroup:test_cpu
       │ cgroup:test_stress.sh
```



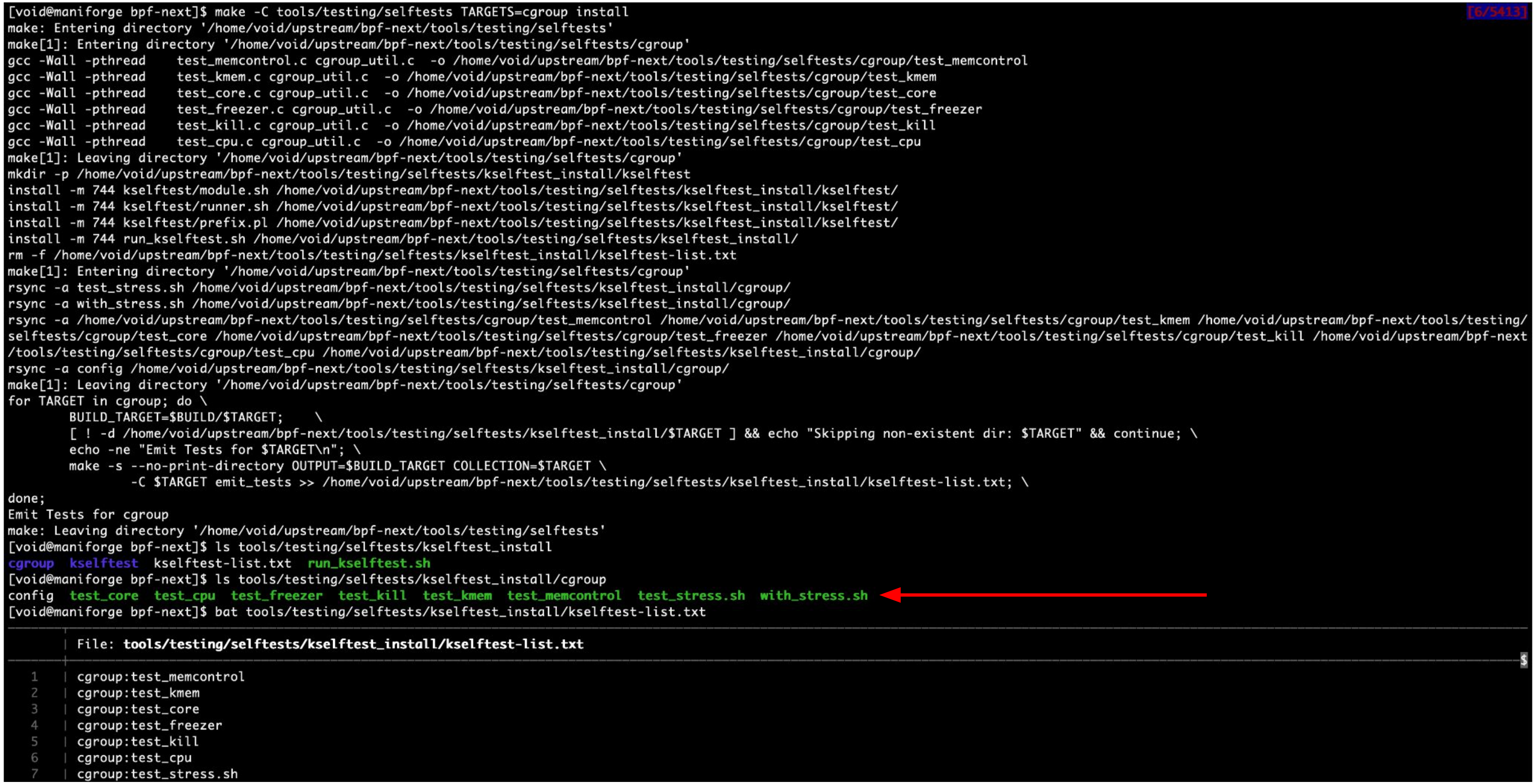
```
[void@maniforge bpf-next]$ make -C tools/testing/selftests TARGETS=cgroup install                                                                                   [6/5413]
make: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests'
make[1]: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
gcc -Wall -pthread    test_memcontrol.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_memcontrol
gcc -Wall -pthread    test_kmem.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kmem
gcc -Wall -pthread    test_core.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_core
gcc -Wall -pthread    test_freezer.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_freezer
gcc -Wall -pthread    test_kill.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kill
gcc -Wall -pthread    test_cpu.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_cpu
make[1]: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
mkdir -p /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest
install -m 744 kselftest/module.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 kselftest/runner.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 kselftest/prefix.pl /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 run_kselftest.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/
rm -f /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest-list.txt
make[1]: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
rsync -a test_stress.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a with_stress.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_memcontrol /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kmem /home/void/upstream/bpf-next/tools/testing/
selftests/cgroup/test_core /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_freezer /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kill /home/void/upstream/bpf-next
/tools/testing/selftests/cgroup/test_cpu /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a config /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
make[1]: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
for TARGET in cgroup; do \
        BUILD_TARGET=$BUILD/$TARGET;    \
        [ ! -d /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/$TARGET ] && echo "Skipping non-existent dir: $TARGET" && continue; \
        echo -ne "Emit Tests for $TARGET\n"; \
        make -s --no-print-directory OUTPUT=$BUILD_TARGET COLLECTION=$TARGET \
                -C $TARGET emit_tests >> /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest-list.txt; \
done;
Emit Tests for cgroup
make: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests'
[void@maniforge bpf-next]$ ls tools/testing/selftests/kselftest_install
cgroup  kselftest  kselftest-list.txt  run_kselftest.sh
[void@maniforge bpf-next]$ ls tools/testing/selftests/kselftest_install/cgroup
config  test_core  test_cpu  test_freezer  test_kill  test_kmem  test_memcontrol  test_stress.sh  with_stress.sh
[void@maniforge bpf-next]$ bat tools/testing/selftests/kselftest_install/kselftest-list.txt
───────┬────────────────────────────────────────────────────────────────────
       │ File: tools/testing/selftests/kselftest_install/kselftest-list.txt
───────┼────────────────────────────────────────────────────────────────────
       │ cgroup:test_memcontrol
       │ cgroup:test_kmem
       │ cgroup:test_core
       │ cgroup:test_freezer
       │ cgroup:test_kill
       │ cgroup:test_cpu
       │ cgroup:test_stress.sh
```



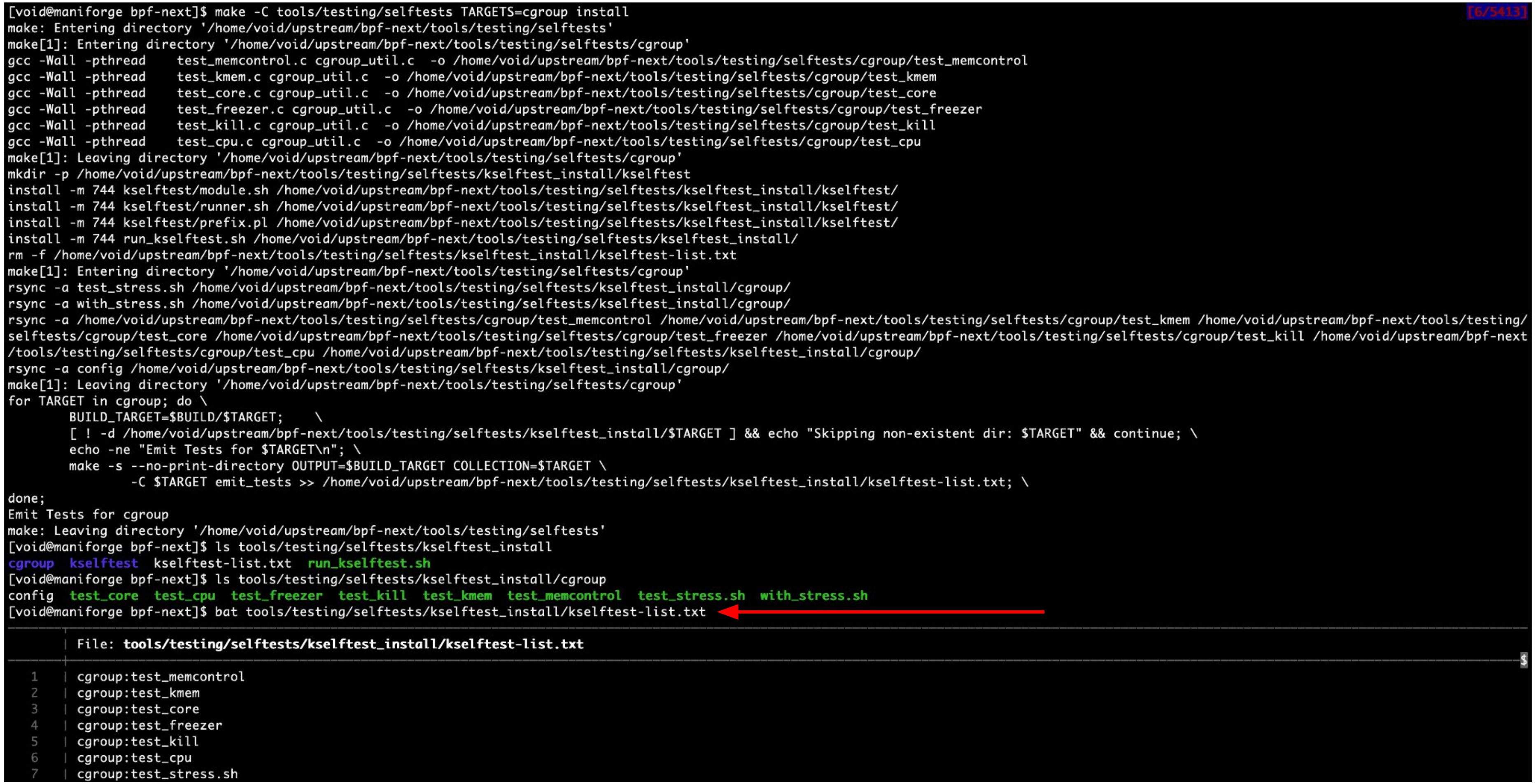
```
[void@maniforge bpf-next]$ make -C tools/testing/selftests TARGETS=cgroup install                                        [6/5413]
make: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests'
make[1]: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
gcc -Wall -pthread    test_memcontrol.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_memcontrol
gcc -Wall -pthread    test_kmem.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kmem
gcc -Wall -pthread    test_core.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_core
gcc -Wall -pthread    test_freezer.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_freezer
gcc -Wall -pthread    test_kill.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kill
gcc -Wall -pthread    test_cpu.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_cpu
make[1]: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
mkdir -p /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest
install -m 744 kselftest/module.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 kselftest/runner.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 kselftest/prefix.pl /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 run_kselftest.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/
rm -f /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest-list.txt
make[1]: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
rsync -a test_stress.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a with_stress.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_memcontrol /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kmem /home/void/upstream/bpf-next/tools/testing/
selftests/cgroup/test_core /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_freezer /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kill /home/void/upstream/bpf-next
/tools/testing/selftests/cgroup/test_cpu /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a config /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
make[1]: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
for TARGET in cgroup; do \
        BUILD_TARGET=$BUILD/$TARGET;    \
        [ ! -d /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/$TARGET ] && echo "Skipping non-existent dir: $TARGET" && continue; \
        echo -ne "Emit Tests for $TARGET\n"; \
        make -s --no-print-directory OUTPUT=$BUILD_TARGET COLLECTION=$TARGET \
                -C $TARGET emit_tests >> /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest-list.txt; \
done;
Emit Tests for cgroup
make: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests'
[void@maniforge bpf-next]$ ls tools/testing/selftests/kselftest_install
cgroup  kselftest  kselftest-list.txt  run_kselftest.sh
[void@maniforge bpf-next]$ ls tools/testing/selftests/kselftest_install/cgroup
config  test_core  test_cpu  test_freezer  test_kill  test_kmem  test_memcontrol  test_stress.sh  with_stress.sh
[void@maniforge bpf-next]$ bat tools/testing/selftests/kselftest_install/kselftest-list.txt
───────┬──────────────────────────────────────────────────────────────
       │ File: tools/testing/selftests/kselftest_install/kselftest-list.txt
───────┼──────────────────────────────────────────────────────────────
       │ cgroup:test_memcontrol
       │ cgroup:test_kmem
       │ cgroup:test_core
       │ cgroup:test_freezer
       │ cgroup:test_kill
       │ cgroup:test_cpu
       │ cgroup:test_stress.sh
```



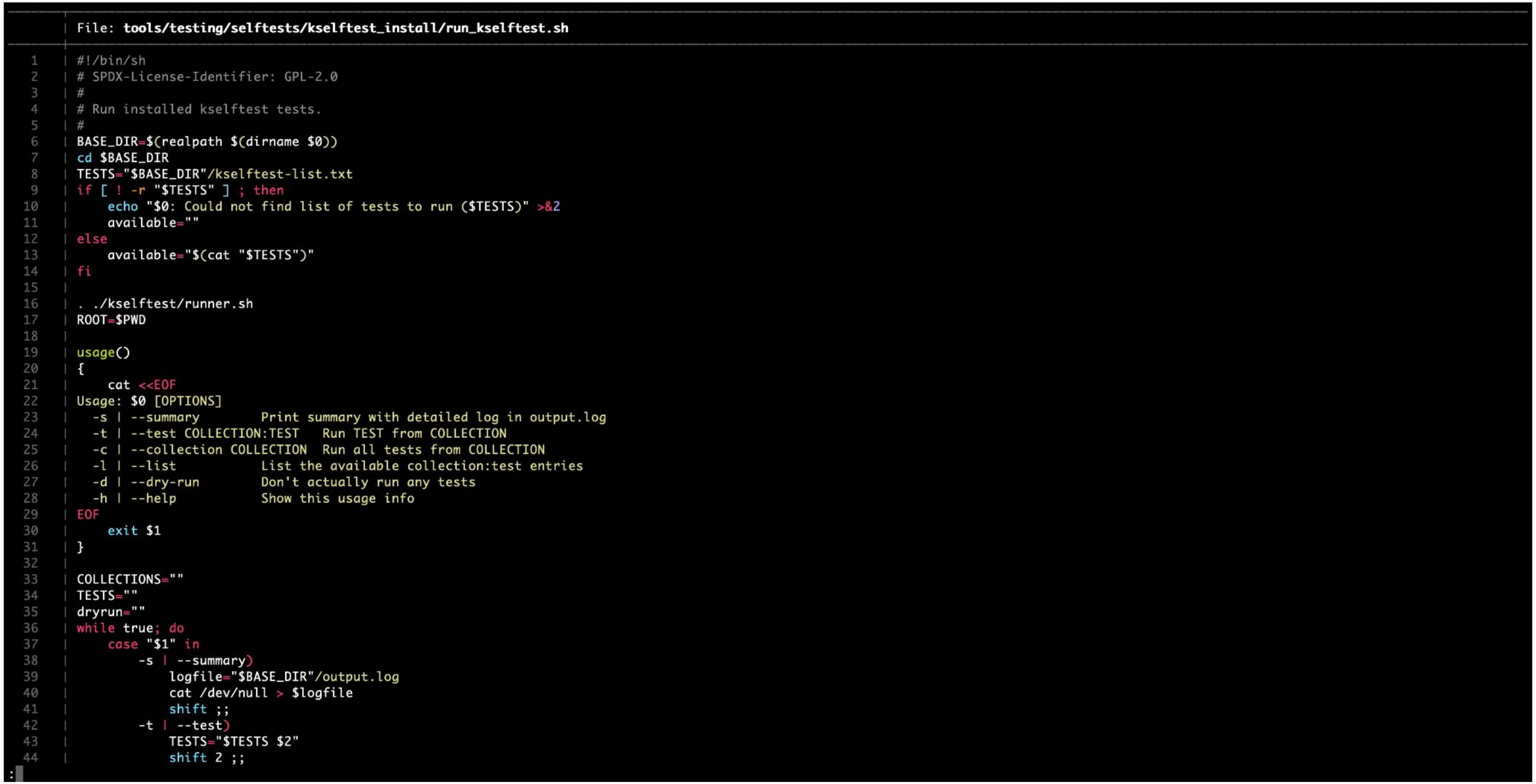
```
[void@maniforge bpf-next]$ make -C tools/testing/selftests TARGETS=cgroup install                                                                                                          [6/5413]
make: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests'
make[1]: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
gcc -Wall -pthread    test_memcontrol.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_memcontrol
gcc -Wall -pthread    test_kmem.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kmem
gcc -Wall -pthread    test_core.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_core
gcc -Wall -pthread    test_freezer.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_freezer
gcc -Wall -pthread    test_kill.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kill
gcc -Wall -pthread    test_cpu.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_cpu
make[1]: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
mkdir -p /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest
install -m 744 kselftest/module.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 kselftest/runner.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 kselftest/prefix.pl /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 run_kselftest.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/
rm -f /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest-list.txt
make[1]: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
rsync -a test_stress.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a with_stress.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_memcontrol /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kmem /home/void/upstream/bpf-next/tools/testing/
selftests/cgroup/test_core /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_freezer /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kill /home/void/upstream/bpf-next
/tools/testing/selftests/cgroup/test_cpu /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a config /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
make[1]: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
for TARGET in cgroup; do \
        BUILD_TARGET=$BUILD/$TARGET;    \
        [ ! -d /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/$TARGET ] && echo "Skipping non-existent dir: $TARGET" && continue; \
        echo -ne "Emit Tests for $TARGET\n"; \
        make -s --no-print-directory OUTPUT=$BUILD_TARGET COLLECTION=$TARGET \
                -C $TARGET emit_tests >> /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest-list.txt; \
done;
Emit Tests for cgroup
make: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests'
[void@maniforge bpf-next]$ ls tools/testing/selftests/kselftest_install
cgroup  kselftest  kselftest-list.txt  run_kselftest.sh
[void@maniforge bpf-next]$ ls tools/testing/selftests/kselftest_install/cgroup
config  test_core  test_cpu  test_freezer  test_kill  test_kmem  test_memcontrol  test_stress.sh  with_stress.sh
[void@maniforge bpf-next]$ bat tools/testing/selftests/kselftest_install/kselftest-list.txt
───────┬────────────────────────────────────────────────────────────────
       │ File: tools/testing/selftests/kselftest_install/kselftest-list.txt
───────┼────────────────────────────────────────────────────────────────
       │ cgroup:test_memcontrol
       │ cgroup:test_kmem
       │ cgroup:test_core
       │ cgroup:test_freezer
       │ cgroup:test_kill
       │ cgroup:test_cpu
       │ cgroup:test_stress.sh
```



```
[void@maniforge bpf-next]$ make -C tools/testing/selftests TARGETS=cgroup install                                          [6/5413]
make: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests'
make[1]: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
gcc -Wall -pthread    test_memcontrol.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_memcontrol
gcc -Wall -pthread    test_kmem.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kmem
gcc -Wall -pthread    test_core.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_core
gcc -Wall -pthread    test_freezer.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_freezer
gcc -Wall -pthread    test_kill.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kill
gcc -Wall -pthread    test_cpu.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_cpu
make[1]: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
mkdir -p /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest
install -m 744 kselftest/module.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 kselftest/runner.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 kselftest/prefix.pl /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 run_kselftest.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/
rm -f /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest-list.txt
make[1]: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
rsync -a test_stress.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a with_stress.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_memcontrol /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kmem /home/void/upstream/bpf-next/tools/testing/
selftests/cgroup/test_core /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_freezer /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kill /home/void/upstream/bpf-next
/tools/testing/selftests/cgroup/test_cpu /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a config /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
make[1]: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
for TARGET in cgroup; do \
        BUILD_TARGET=$BUILD/$TARGET;    \
        [ ! -d /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/$TARGET ] && echo "Skipping non-existent dir: $TARGET" && continue; \
        echo -ne "Emit Tests for $TARGET\n"; \
        make -s --no-print-directory OUTPUT=$BUILD_TARGET COLLECTION=$TARGET \
                -C $TARGET emit_tests >> /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest-list.txt; \
done;
Emit Tests for cgroup
make: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests'
[void@maniforge bpf-next]$ ls tools/testing/selftests/kselftest_install
cgroup  kselftest  kselftest-list.txt  run_kselftest.sh
[void@maniforge bpf-next]$ ls tools/testing/selftests/kselftest_install/cgroup
config  test_core  test_cpu  test_freezer  test_kill  test_kmem  test_memcontrol  test_stress.sh  with_stress.sh
[void@maniforge bpf-next]$ bat tools/testing/selftests/kselftest_install/kselftest-list.txt
───────┬────────────────────────────────────────────────────────────────────────────────
       │ File: tools/testing/selftests/kselftest_install/kselftest-list.txt
───────┼────────────────────────────────────────────────────────────────────────────────
       │ cgroup:test_memcontrol
       │ cgroup:test_kmem
       │ cgroup:test_core
       │ cgroup:test_freezer
       │ cgroup:test_kill
       │ cgroup:test_cpu
       │ cgroup:test_stress.sh
```



```
[void@maniforge bpf-next]$ make -C tools/testing/selftests TARGETS=cgroup install                                                                    [6/5413]
make: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests'
make[1]: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
gcc -Wall -pthread    test_memcontrol.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_memcontrol
gcc -Wall -pthread    test_kmem.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kmem
gcc -Wall -pthread    test_core.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_core
gcc -Wall -pthread    test_freezer.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_freezer
gcc -Wall -pthread    test_kill.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kill
gcc -Wall -pthread    test_cpu.c cgroup_util.c  -o /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_cpu
make[1]: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
mkdir -p /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest
install -m 744 kselftest/module.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 kselftest/runner.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 kselftest/prefix.pl /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest/
install -m 744 run_kselftest.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/
rm -f /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest-list.txt
make[1]: Entering directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
rsync -a test_stress.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a with_stress.sh /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_memcontrol /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kmem /home/void/upstream/bpf-next/tools/testing/
selftests/cgroup/test_core /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_freezer /home/void/upstream/bpf-next/tools/testing/selftests/cgroup/test_kill /home/void/upstream/bpf-next
/tools/testing/selftests/cgroup/test_cpu /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
rsync -a config /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/cgroup/
make[1]: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests/cgroup'
for TARGET in cgroup; do \
        BUILD_TARGET=$BUILD/$TARGET;    \
        [ ! -d /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/$TARGET ] && echo "Skipping non-existent dir: $TARGET" && continue; \
        echo -ne "Emit Tests for $TARGET\n"; \
        make -s --no-print-directory OUTPUT=$BUILD_TARGET COLLECTION=$TARGET \
                -C $TARGET emit_tests >> /home/void/upstream/bpf-next/tools/testing/selftests/kselftest_install/kselftest-list.txt; \
done;
Emit Tests for cgroup
make: Leaving directory '/home/void/upstream/bpf-next/tools/testing/selftests'
[void@maniforge bpf-next]$ ls tools/testing/selftests/kselftest_install
cgroup  kselftest  kselftest-list.txt  run_kselftest.sh
[void@maniforge bpf-next]$ ls tools/testing/selftests/kselftest_install/cgroup
config  test_core  test_cpu  test_freezer  test_kill  test_kmem  test_memcontrol  test_stress.sh  with_stress.sh
[void@maniforge bpf-next]$ bat tools/testing/selftests/kselftest_install/kselftest-list.txt
───────┬────────────────────────────────────────────────────────────────
       │ File: tools/testing/selftests/kselftest_install/kselftest-list.txt
───────┼────────────────────────────────────────────────────────────────
       │ cgroup:test_memcontrol
       │ cgroup:test_kmem
       │ cgroup:test_core
       │ cgroup:test_freezer
       │ cgroup:test_kill
       │ cgroup:test_cpu
       │ cgroup:test_stress.sh
```

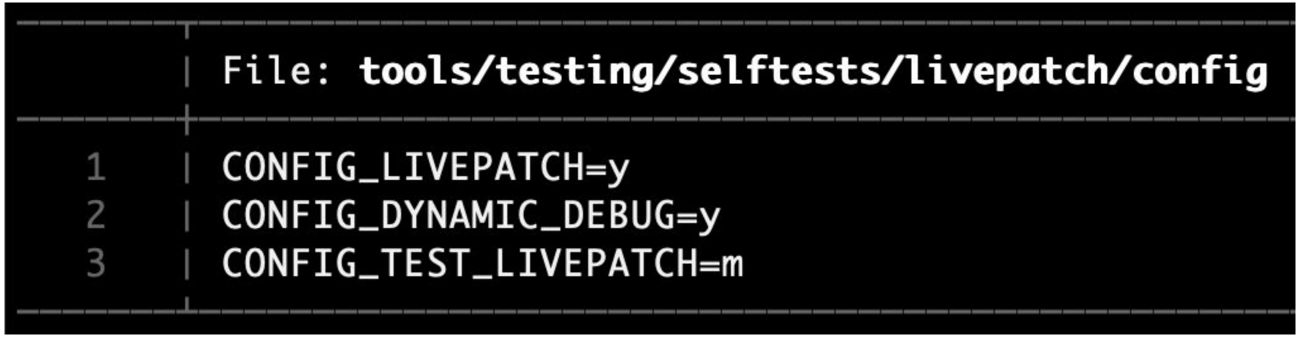


```
File: tools/testing/selftests/kselftest_install/run_kselftest.sh
```

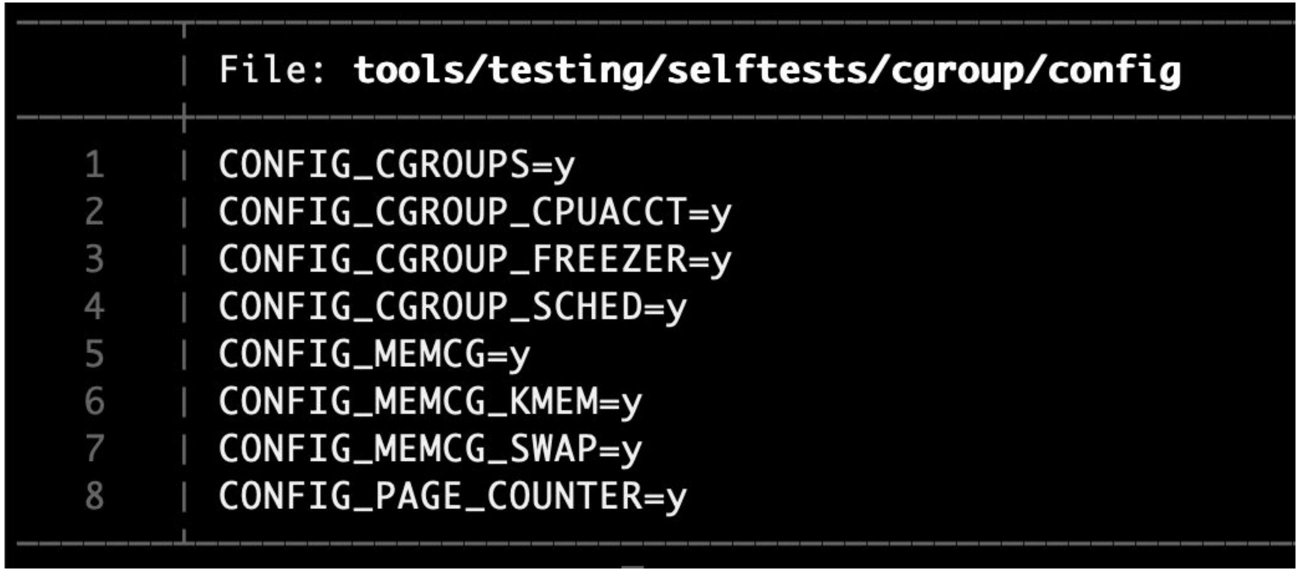
```sh
#!/bin/sh
# SPDX-License-Identifier: GPL-2.0
#
# Run installed kselftest tests.
#
BASE_DIR=$(realpath $(dirname $0))
cd $BASE_DIR
TESTS="$BASE_DIR"/kselftest-list.txt
if [ ! -r "$TESTS" ] ; then
    echo "$0: Could not find list of tests to run ($TESTS)" >&2
    available=""
else
    available="$(cat "$TESTS")"
fi

. ./kselftest/runner.sh
ROOT=$PWD

usage()
{
    cat <<EOF
Usage: $0 [OPTIONS]
  -s | --summary        Print summary with detailed log in output.log
  -t | --test COLLECTION:TEST   Run TEST from COLLECTION
  -c | --collection COLLECTION  Run all tests from COLLECTION
  -l | --list           List the available collection:test entries
  -d | --dry-run        Don't actually run any tests
  -h | --help           Show this usage info
EOF
    exit $1
}

COLLECTIONS=""
TESTS=""
dryrun=""
while true; do
    case "$1" in
        -s | --summary)
            logfile="$BASE_DIR"/output.log
            cat /dev/null > $logfile
            shift ;;
        -t | --test)
            TESTS="$TESTS $2"
            shift 2 ;;
:
```

A kselftest suite can specify requisite kconfig options

- Test-suite advertising which Kconfig options it requires to run
- Not actually relevant to the building or packaging of kselftests
  - Can run kselftests on a kernel that does not have Kconfig options
  - Really just present by convention



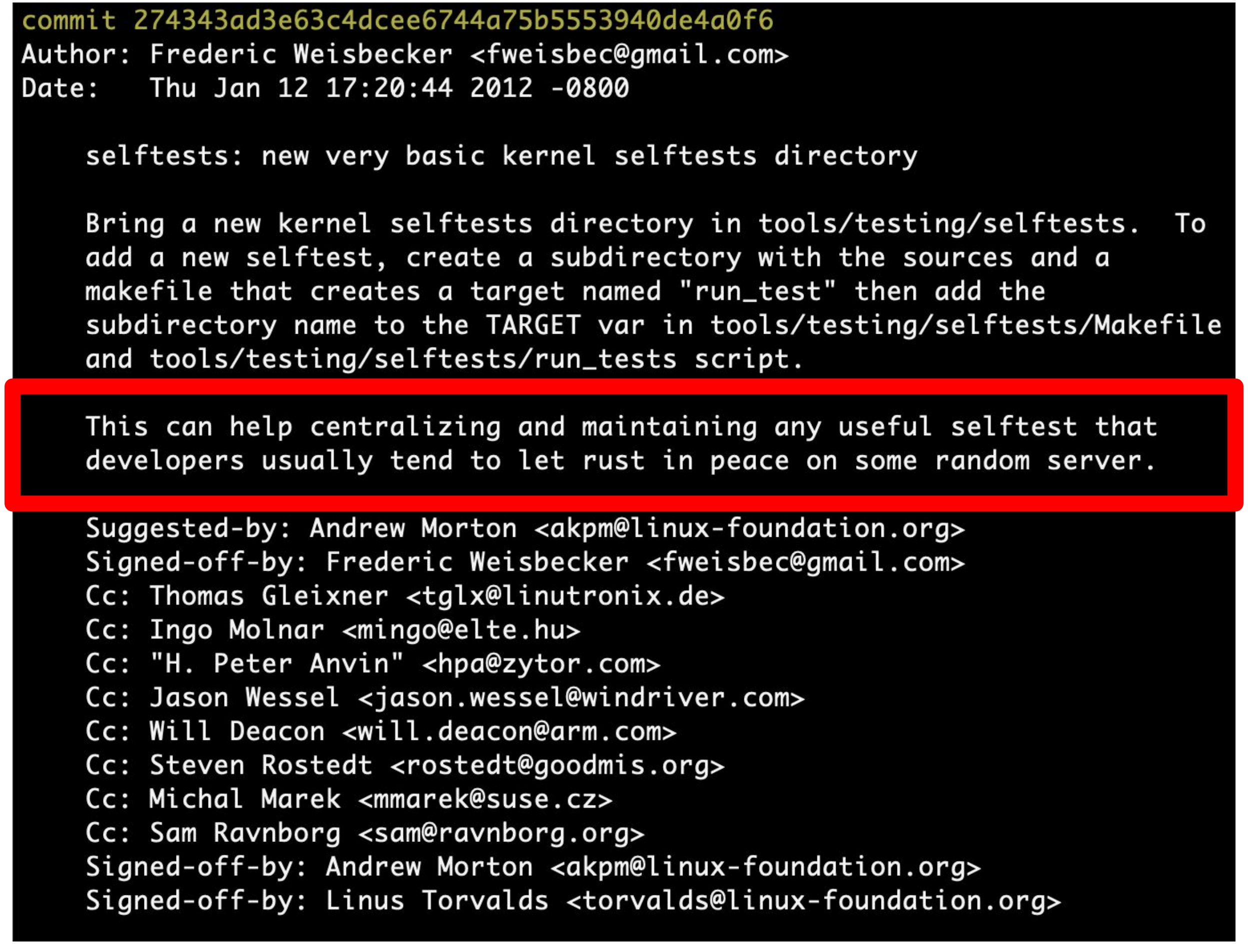
```
File: tools/testing/selftests/livepatch/config

CONFIG_LIVEPATCH=y
CONFIG_DYNAMIC_DEBUG=y
CONFIG_TEST_LIVEPATCH=m
```



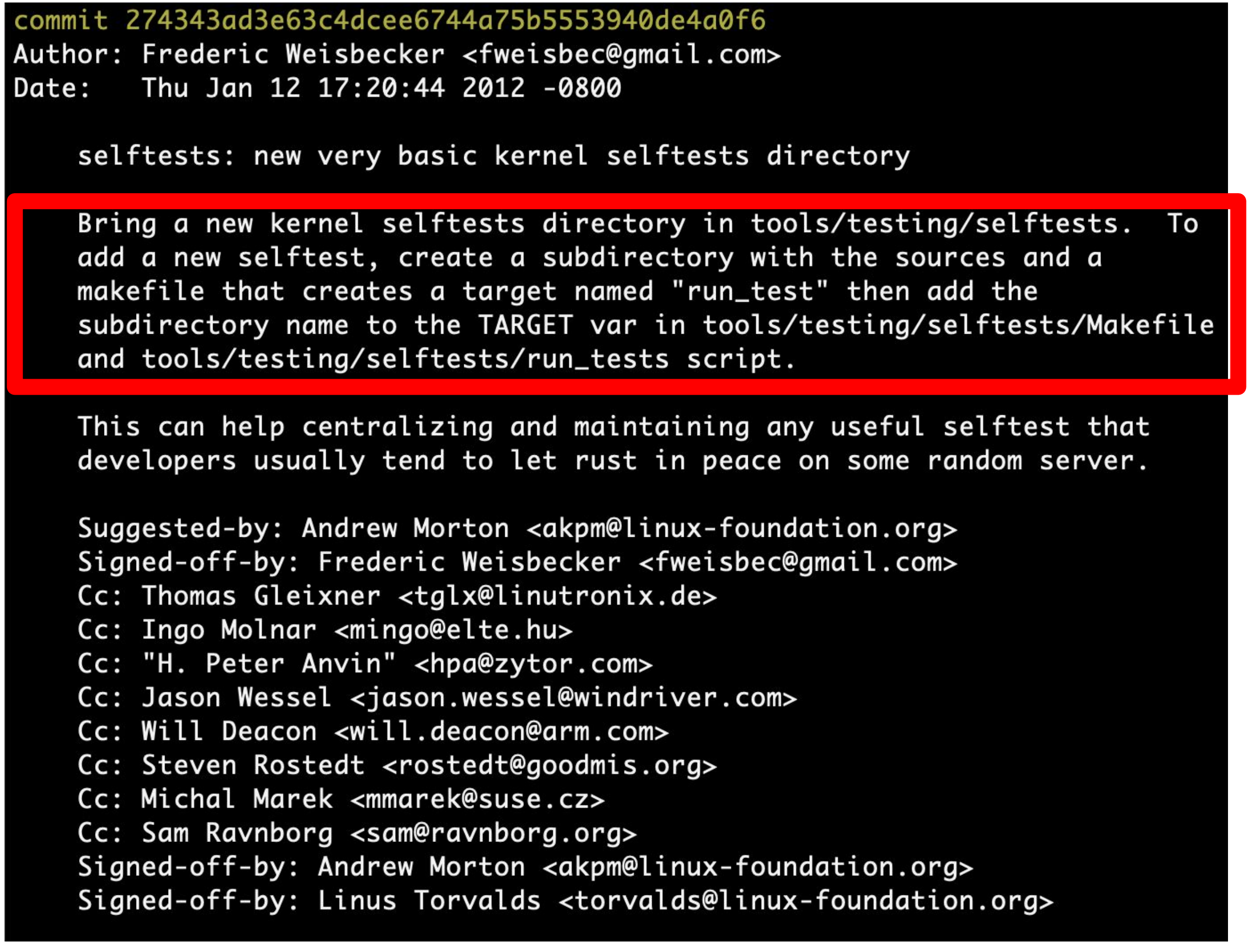
```
File: tools/testing/selftests/cgroup/config

CONFIG_CGROUPS=y
CONFIG_CGROUP_CPUACCT=y
CONFIG_CGROUP_FREEZER=y
CONFIG_CGROUP_SCHED=y
CONFIG_MEMCG=y
CONFIG_MEMCG_KMEM=y
CONFIG_MEMCG_SWAP=y
CONFIG_PAGE_COUNTER=y
```

Note: Lots of discussion expected (and hoped for) during this section. Please feel free to interject.

kselftest was designed for ad-hoc usage



```
commit 274343ad3e63c4dcee6744a75b5553940de4a0f6
Author: Frederic Weisbecker <fweisbec@gmail.com>
Date:   Thu Jan 12 17:20:44 2012 -0800

    selftests: new very basic kernel selftests directory

    Bring a new kernel selftests directory in tools/testing/selftests.  To
    add a new selftest, create a subdirectory with the sources and a
    makefile that creates a target named "run_test" then add the
    subdirectory name to the TARGET var in tools/testing/selftests/Makefile
    and tools/testing/selftests/run_tests script.

    This can help centralizing and maintaining any useful selftest that
    developers usually tend to let rust in peace on some random server.

    Suggested-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Frederic Weisbecker <fweisbec@gmail.com>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Ingo Molnar <mingo@elte.hu>
    Cc: "H. Peter Anvin" <hpa@zytor.com>
    Cc: Jason Wessel <jason.wessel@windriver.com>
    Cc: Will Deacon <will.deacon@arm.com>
    Cc: Steven Rostedt <rostedt@goodmis.org>
    Cc: Michal Marek <mmarek@suse.cz>
    Cc: Sam Ravnborg <sam@ravnborg.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
```



```
commit 274343ad3e63c4dcee6744a75b5553940de4a0f6
Author: Frederic Weisbecker <fweisbec@gmail.com>
Date:   Thu Jan 12 17:20:44 2012 -0800

    selftests: new very basic kernel selftests directory

    Bring a new kernel selftests directory in tools/testing/selftests.  To
    add a new selftest, create a subdirectory with the sources and a
    makefile that creates a target named "run_test" then add the
    subdirectory name to the TARGET var in tools/testing/selftests/Makefile
    and tools/testing/selftests/run_tests script.

    This can help centralizing and maintaining any useful selftest that
    developers usually tend to let rust in peace on some random server.

    Suggested-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Frederic Weisbecker <fweisbec@gmail.com>
    Cc: Thomas Gleixner <tglx@linutronix.de>
    Cc: Ingo Molnar <mingo@elte.hu>
    Cc: "H. Peter Anvin" <hpa@zytor.com>
    Cc: Jason Wessel <jason.wessel@windriver.com>
    Cc: Will Deacon <will.deacon@arm.com>
    Cc: Steven Rostedt <rostedt@goodmis.org>
    Cc: Michal Marek <mmarek@suse.cz>
    Cc: Sam Ravnborg <sam@ravnborg.org>
    Signed-off-by: Andrew Morton <akpm@linux-foundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
```

Since then, kselftests has some new features:

- Test suites can be packaged and installed
- Test runner can run test cases, parse KTAP output
- More built-in Make variables supported

But no common expectation for configuration

- config file is by convention, not used when packaging
- Test runner only runs executables and parses output, no build automation, VM spawning, etc
- Many test suites don’t use TEST_PROGS, TEST_GEN_PROGS

What’s the long-term roadmap for kselftest?

- New features being added to kselftest makes it more like a full-fledged testing framework
- At this point, seems to have two responsibilities:

1. House test-code that is specific to each subsystem, and structured to the liking of maintainers (original)
2. Provide a framework for defining, building, and running tests (new)

Assuming kselftest should evolve, what should it do?

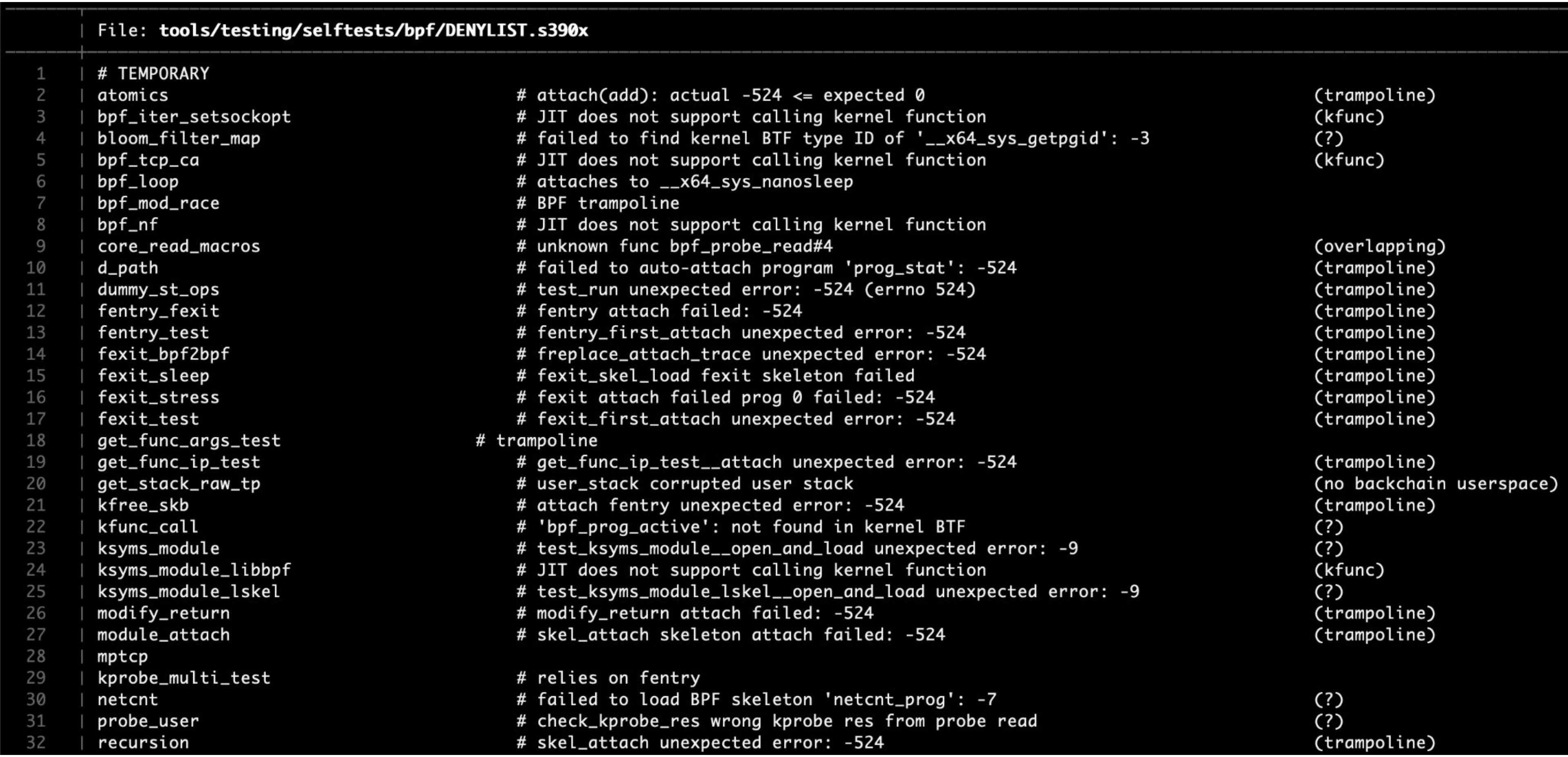
- Should kselftest become a more fully-featured testing framework?
- Should it dictate more structure to test suites and test cases?
- Should the test runner do more for users?

A single config file may not be enough

- Should we standardize how test suites structure themselves?
  - Globally required configs
  - Arch-specific configs
  - Per-testcase configs?
- Some configurations are mutually exclusive
  - E.g. CONFIG_ARM64 and CONFIG_X86_64
- Some features may only be available on certain architectures

Some test suites have already defined this for themselves

- BPF has a DENYLIST.s390x file which signals to CI which testcases aren’t supported on s390x
- Also has a global DENYLIST for signaling which testcases are broken and should be ignored



```
File: tools/testing/selftests/bpf/DENYLIST.s390x

# TEMPORARY
atomics                                  # attach(add): actual -524 <= expected 0                                       (trampoline)
bpf_iter_setsockopt                      # JIT does not support calling kernel function                                 (kfunc)
bloom_filter_map                         # failed to find kernel BTF type ID of '__x64_sys_getpgid': -3                 (?)
bpf_tcp_ca                               # JIT does not support calling kernel function                                 (kfunc)
bpf_loop                                 # attaches to __x64_sys_nanosleep
bpf_mod_race                             # BPF trampoline
bpf_nf                                   # JIT does not support calling kernel function
core_read_macros                         # unknown func bpf_probe_read#4                                                (overlapping)
d_path                                   # failed to auto-attach program 'prog_stat': -524                              (trampoline)
dummy_st_ops                             # test_run unexpected error: -524 (errno 524)                                  (trampoline)
fentry_fexit                             # fentry attach failed: -524                                                   (trampoline)
fentry_test                              # fentry_first_attach unexpected error: -524                                   (trampoline)
fexit_bpf2bpf                            # freplace_attach_trace unexpected error: -524                                 (trampoline)
fexit_sleep                              # fexit_skel_load fexit skeleton failed                                        (trampoline)
fexit_stress                             # fexit attach failed prog 0 failed: -524                                      (trampoline)
fexit_test                               # fexit_first_attach unexpected error: -524                                    (trampoline)
get_func_args_test                   # trampoline
get_func_ip_test                         # get_func_ip_test__attach unexpected error: -524                              (trampoline)
get_stack_raw_tp                         # user_stack corrupted user stack                                              (no backchain userspace)
kfree_skb                                # attach fentry unexpected error: -524                                         (trampoline)
kfunc_call                               # 'bpf_prog_active': not found in kernel BTF                                   (?)
ksyms_module                             # test_ksyms_module__open_and_load unexpected error: -9                        (?)
ksyms_module_libbpf                      # JIT does not support calling kernel function                                 (kfunc)
ksyms_module_lskel                       # test_ksyms_module_lskel__open_and_load unexpected error: -9                  (?)
modify_return                            # modify_return attach failed: -524                                            (trampoline)
module_attach                            # skel_attach skeleton attach failed: -524                                     (trampoline)
mptcp
kprobe_multi_test                        # relies on fentry
netcnt                                   # failed to load BPF skeleton 'netcnt_prog': -7                                (?)
probe_user                               # check_kprobe_res wrong kprobe res from probe read                            (?)
recursion                                # skel_attach unexpected error: -524                                           (trampoline)
```

Running tests on a local build is challenging

Depending on the test-suite, requires a few steps (at least for me):

1. Compile kernel with the correct .config options, manually appended from a selftest suite
2. Boot into a VM, with a mounted volume shared from the host
3. Compile and install kselftests into that mounted volume
4. Run the installed kselftests runner from the VM

Should the runner handle some of these steps?

- Builds the kernel for one or more test-suites, assuming no conflicts
- Boot a VM with some # of cores, amount of RAM
    - Will have to be configurable to accommodate tests that require specific I/O interface configurations, etc
- Run the tests in the VM
- Report results back to the user

- Or, is this something that should be handled at a higher level?

Some test suites already do this, e.g. RCU

- tools/testing/selftests/rcutorture/bin/kvm.sh
    - Runs a VM with some specified # of CPUs, memory, initrd, etc
- tools/testing/selftests/rcutorture/bin/kvm-build.sh
    - Builds a Linux kernel that can be booted into a VM for rcutorture tests
- Should this be a service provided by the core kselftest framework?

Kconfig is easy to mess up

- Some config options may conflict with what’s already present in .config. Kconfig may silently override and drop those options
- Can we add make targets that build the kernel for specific kselftest suites?
  - Could be leveraged by CI jobs
  - Can fail and/or warn if there are conflicting config options
  - Can allow the user to specify specific architectures

Pick your poison, there are a few options

- KernelCI (https://foundation.kernelci.org)
- Patchwork + github + extra magic (https://patchwork.kernel.org/project/netdevbpf/list/)

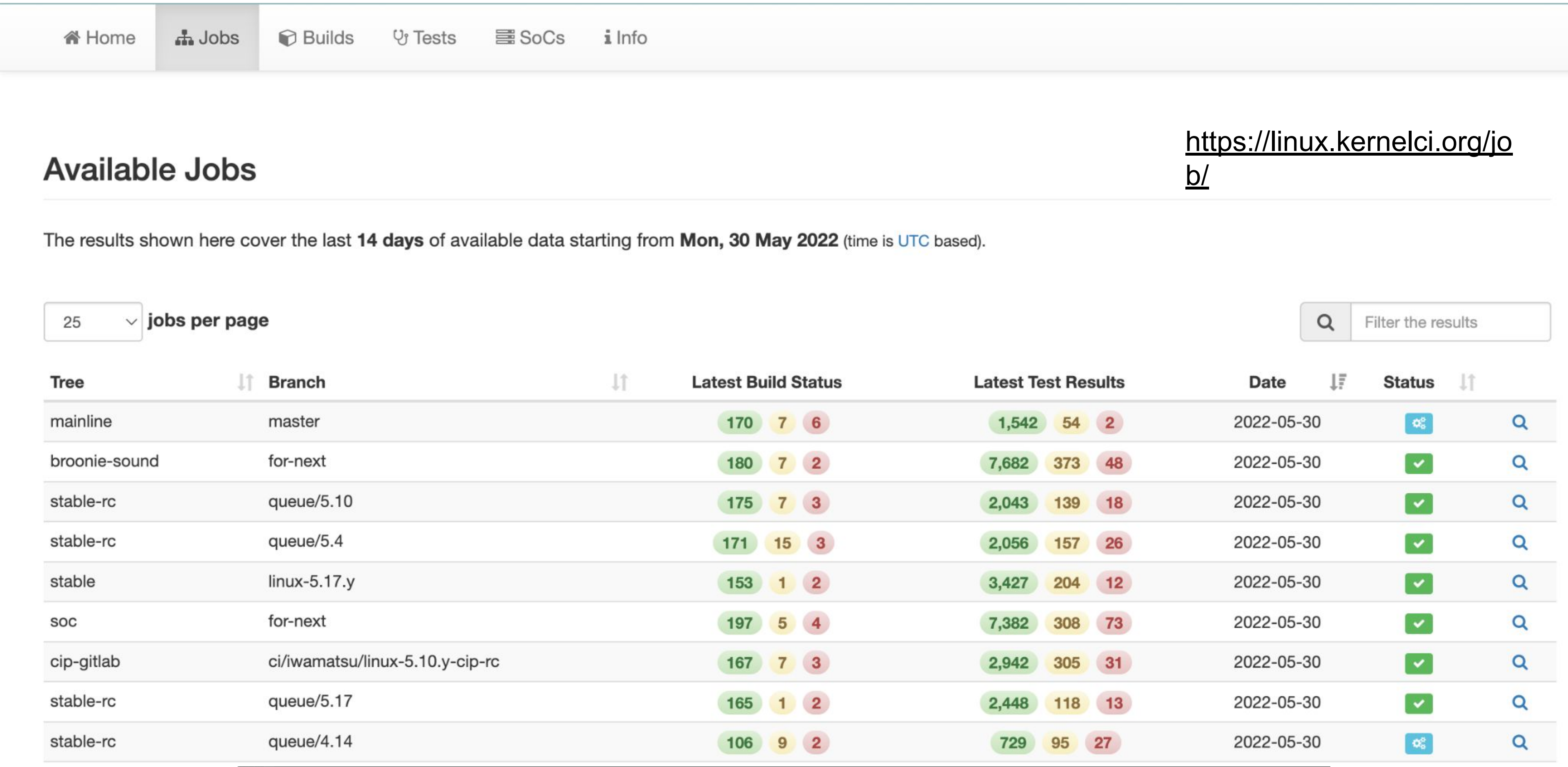
KernelCI

KernelCI – A Linux Foundation project

Open source test automation system

Builds and runs kernels across a variety of trees, branches, toolchains, and configs

Also runs tests on different architectures and SoCs https://linux.kernelci.org/job/



Home Jobs Builds Tests SoCs Info

Available Jobs

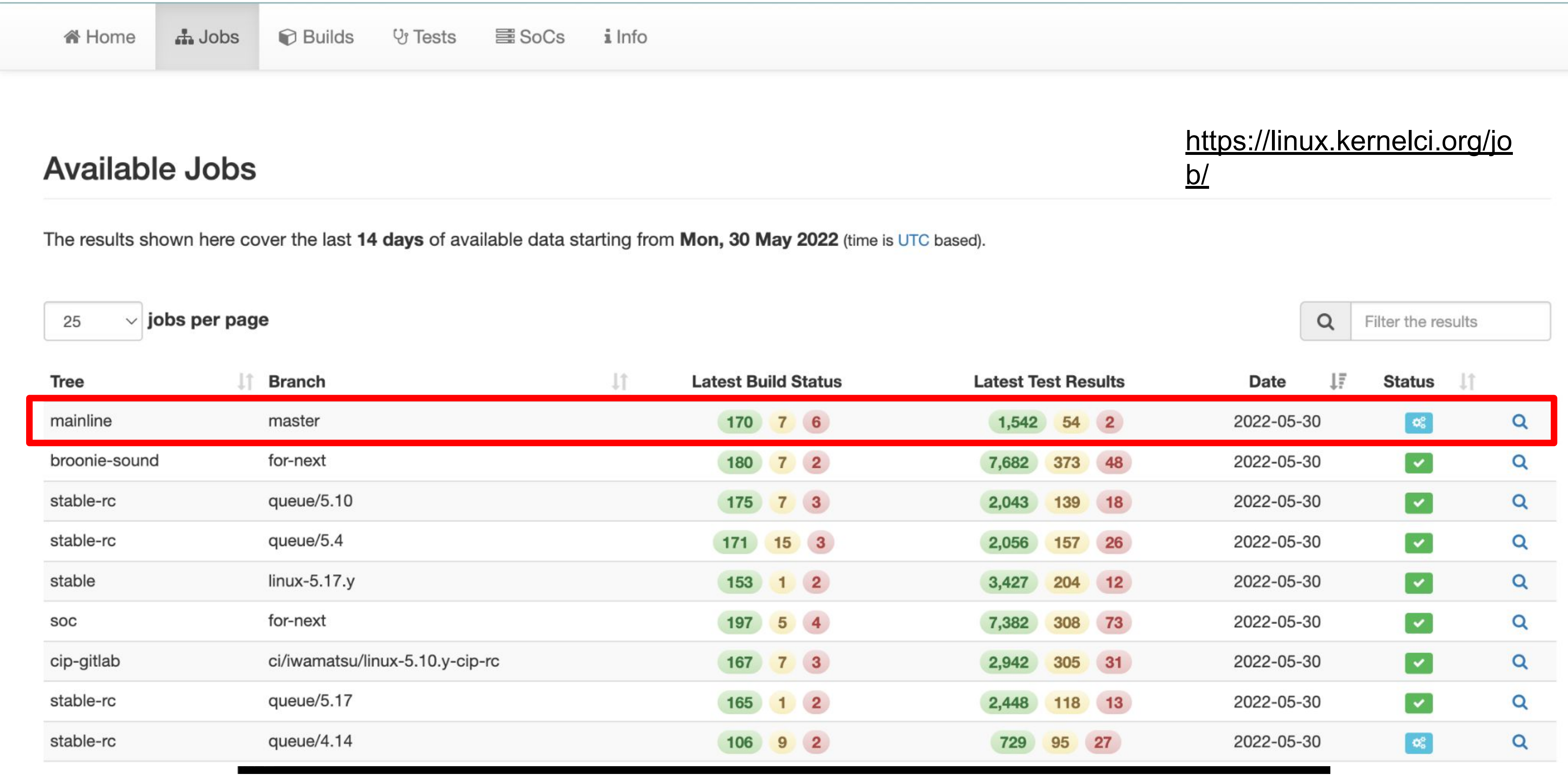
The results shown here cover the last **14 days** of available data starting from **Mon, 30 May 2022** (time is UTC based).

25 **jobs per page**

Filter the results

Tree	Branch	Latest Build Status	Latest Test Results	Date	Status	
mainline	master	170 7 6	1,542 54 2	2022-05-30	⚙	🔍
broonie-sound	for-next	180 7 2	7,682 373 48	2022-05-30	✓	🔍
stable-rc	queue/5.10	175 7 3	2,043 139 18	2022-05-30	✓	🔍
stable-rc	queue/5.4	171 15 3	2,056 157 26	2022-05-30	✓	🔍
stable	linux-5.17.y	153 1 2	3,427 204 12	2022-05-30	✓	🔍
soc	for-next	197 5 4	7,382 308 73	2022-05-30	✓	🔍
cip-gitlab	ci/iwamatsu/linux-5.10.y-cip-rc	167 7 3	2,942 305 31	2022-05-30	✓	🔍
stable-rc	queue/5.17	165 1 2	2,448 118 13	2022-05-30	✓	🔍
stable-rc	queue/4.14	106 9 2	729 95 27	2022-05-30	⚙	🔍

https://linux.kernelci.org/job/



Home | Jobs | Builds | Tests | SoCs | Info

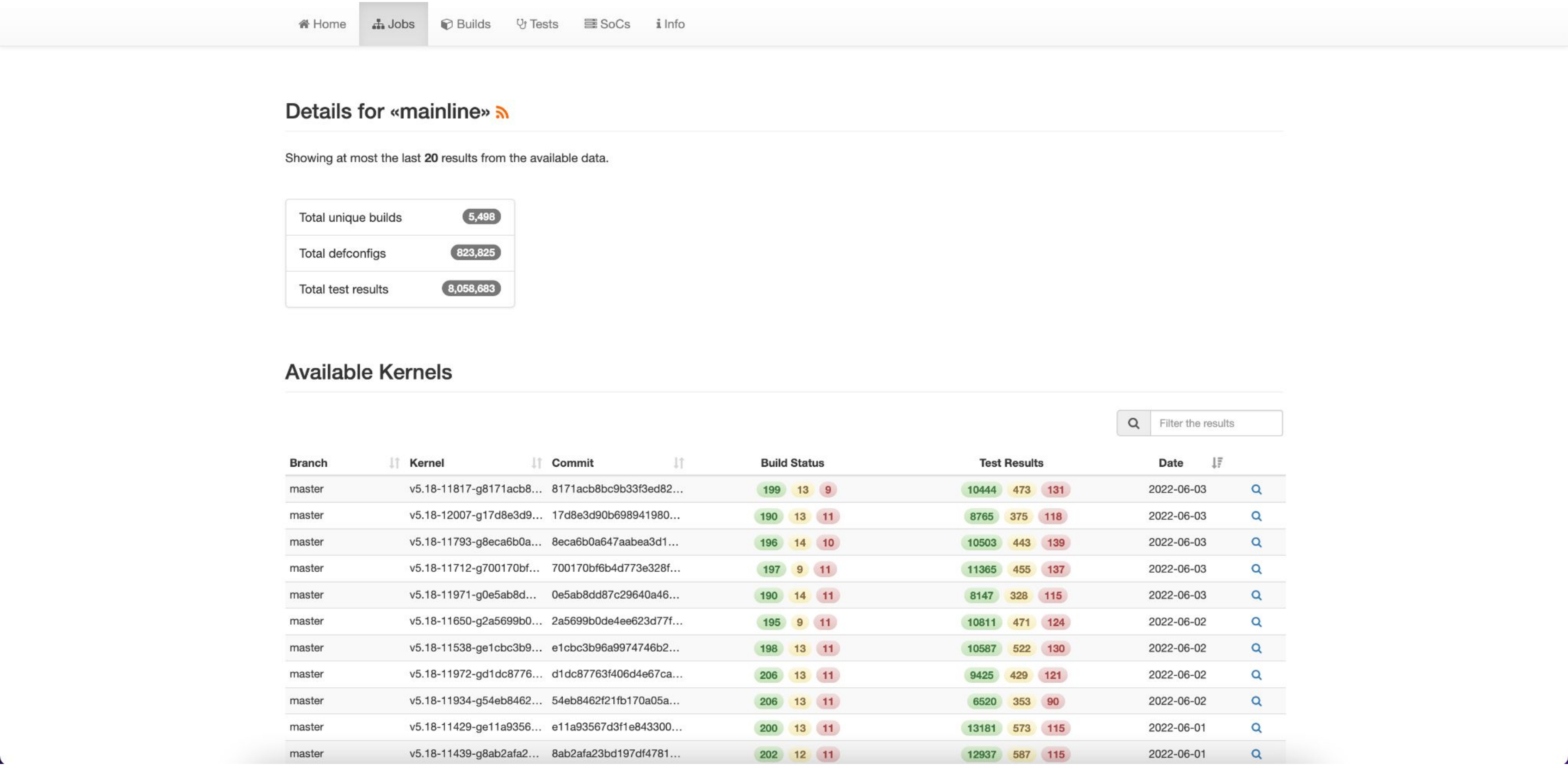
Available Jobs

The results shown here cover the last **14 days** of available data starting from **Mon, 30 May 2022** (time is UTC based).

25 **jobs per page**

Filter the results

Tree	Branch	Latest Build Status	Latest Test Results	Date	Status	
mainline	master	170 7 6	1,542 54 2	2022-05-30	⚙	🔍
broonie-sound	for-next	180 7 2	7,682 373 48	2022-05-30	✓	🔍
stable-rc	queue/5.10	175 7 3	2,043 139 18	2022-05-30	✓	🔍
stable-rc	queue/5.4	171 15 3	2,056 157 26	2022-05-30	✓	🔍
stable	linux-5.17.y	153 1 2	3,427 204 12	2022-05-30	✓	🔍
soc	for-next	197 5 4	7,382 308 73	2022-05-30	✓	🔍
cip-gitlab	ci/iwamatsu/linux-5.10.y-cip-rc	167 7 3	2,942 305 31	2022-05-30	✓	🔍
stable-rc	queue/5.17	165 1 2	2,448 118 13	2022-05-30	✓	🔍
stable-rc	queue/4.14	106 9 2	729 95 27	2022-05-30	⚙	🔍



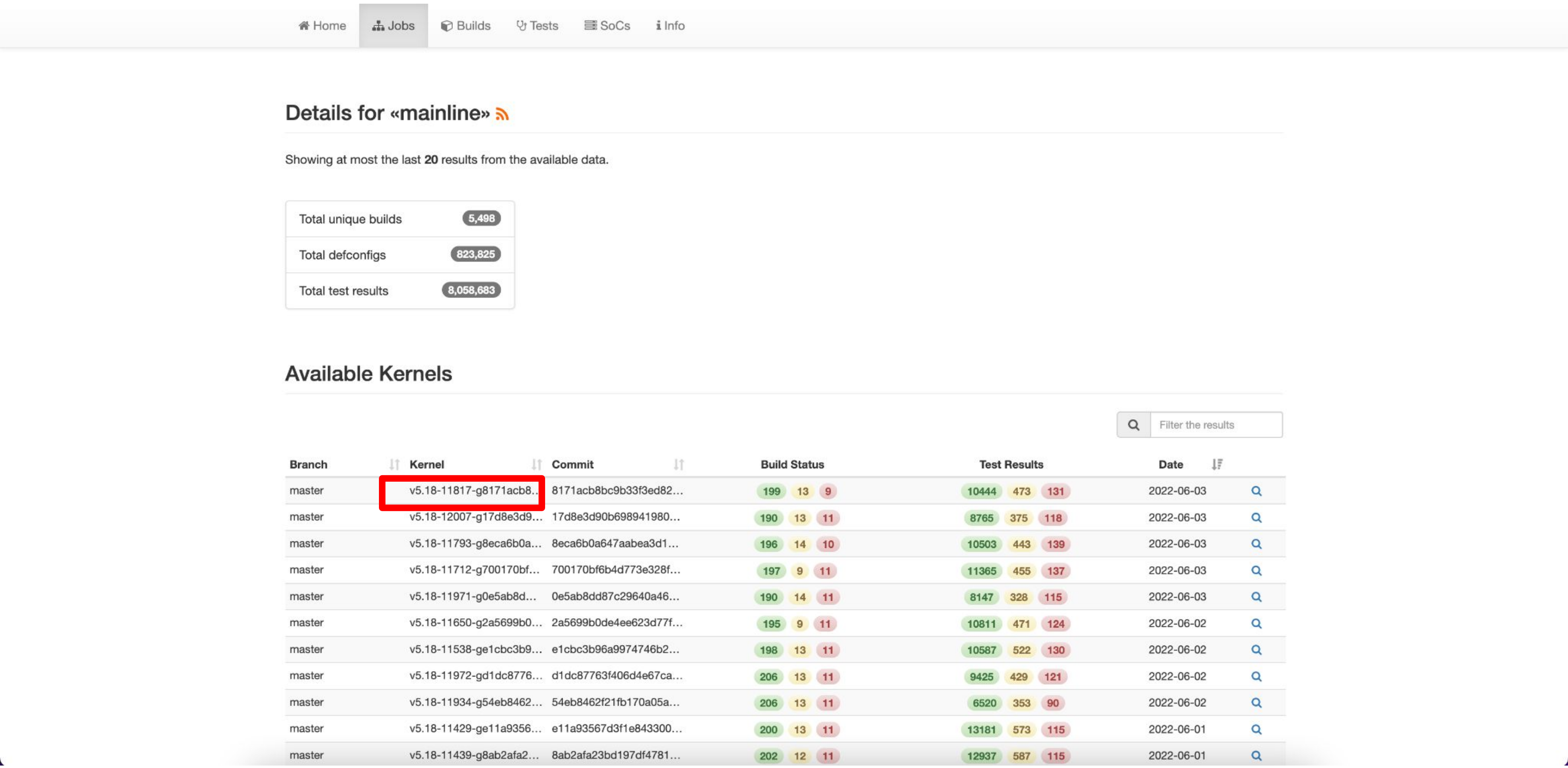
Details for «mainline»

Showing at most the last **20** results from the available data.

Total unique builds	5,498
Total defconfigs	823,825
Total test results	8,058,683

Available Kernels

Branch	Kernel	Commit	Build Status	Test Results	Date
master	v5.18-11817-g8171acb8…	8171acb8bc9b33f3ed82…	199 13 9	10444 473 131	2022-06-03
master	v5.18-12007-g17d8e3d9…	17d8e3d90b698941980…	190 13 11	8765 375 118	2022-06-03
master	v5.18-11793-g8eca6b0a…	8eca6b0a647aabea3d1…	196 14 10	10503 443 139	2022-06-03
master	v5.18-11712-g700170bf…	700170bf6b4d773e328f…	197 9 11	11365 455 137	2022-06-03
master	v5.18-11971-g0e5ab8d…	0e5ab8dd87c29640a46…	190 14 11	8147 328 115	2022-06-03
master	v5.18-11650-g2a5699b0…	2a5699b0de4ee623d77f…	195 9 11	10811 471 124	2022-06-02
master	v5.18-11538-ge1cbc3b9…	e1cbc3b96a9974746b2…	198 13 11	10587 522 130	2022-06-02
master	v5.18-11972-gd1dc8776…	d1dc87763f406d4e67ca…	206 13 11	9425 429 121	2022-06-02
master	v5.18-11934-g54eb8462…	54eb8462f21fb170a05a…	206 13 11	6520 353 90	2022-06-02
master	v5.18-11429-ge11a9356…	e11a93567d3f1e843300…	200 13 11	13181 573 115	2022-06-01
master	v5.18-11439-g8ab2afa2…	8ab2afa23bd197df4781…	202 12 11	12937 587 115	2022-06-01



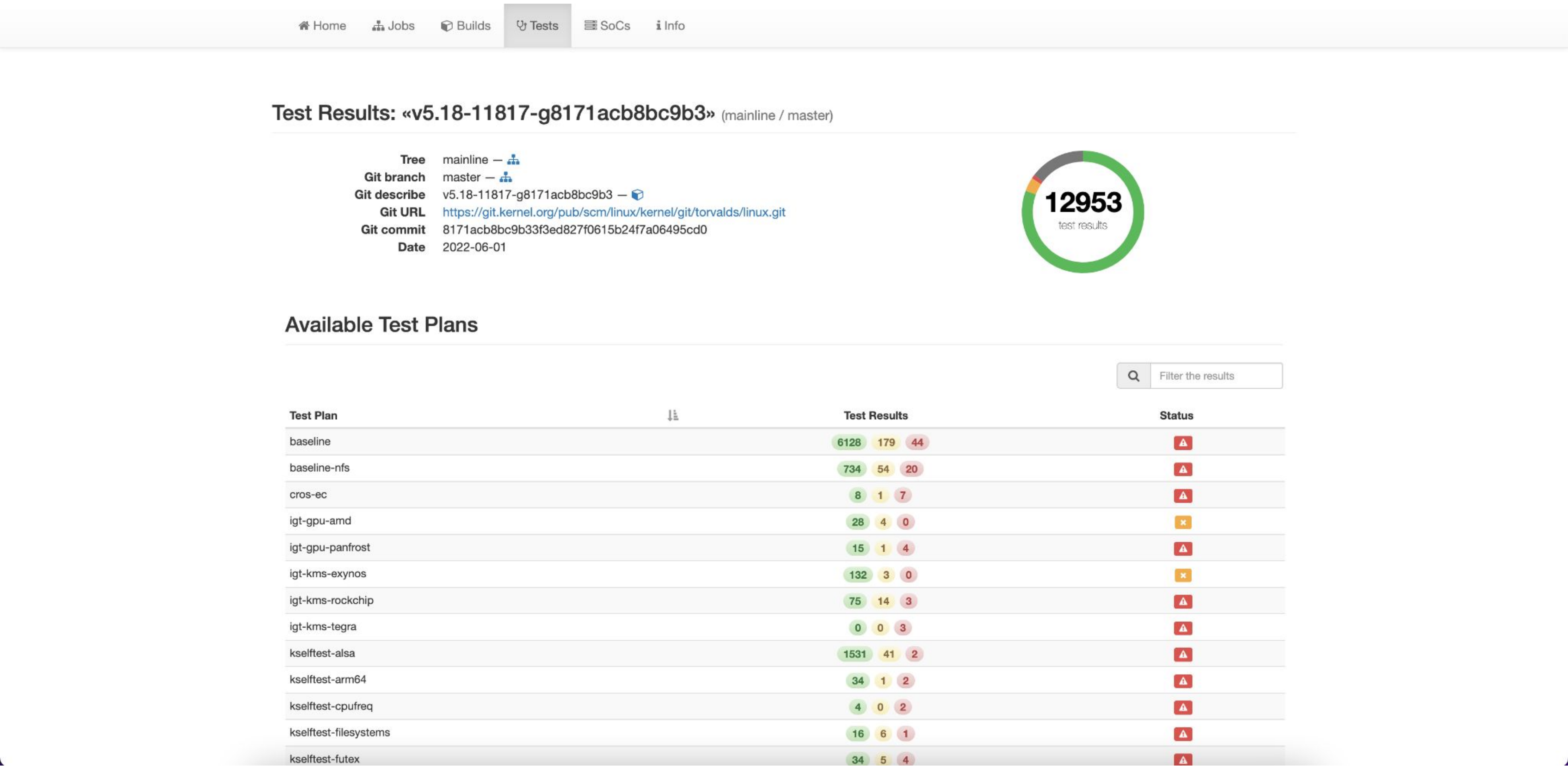
Details for «mainline»

Showing at most the last **20** results from the available data.

Total unique builds	5,498
Total defconfigs	823,825
Total test results	8,058,683

Available Kernels

Branch	Kernel	Commit	Build Status	Test Results	Date
master	v5.18-11817-g8171acb8...	8171acb8bc9b33f3ed82...	199 13 9	10444 473 131	2022-06-03
master	v5.18-12007-g17d8e3d9...	17d8e3d90b698941980...	190 13 11	8765 375 118	2022-06-03
master	v5.18-11793-g8eca6b0a...	8eca6b0a647aabea3d1...	196 14 10	10503 443 139	2022-06-03
master	v5.18-11712-g700170bf...	700170bf6b4d773e328f...	197 9 11	11365 455 137	2022-06-03
master	v5.18-11971-g0e5ab8d...	0e5ab8dd87c29640a46...	190 14 11	8147 328 115	2022-06-03
master	v5.18-11650-g2a5699b0...	2a5699b0de4ee623d77f...	195 9 11	10811 471 124	2022-06-02
master	v5.18-11538-ge1cbc3b9...	e1cbc3b96a9974746b2...	198 13 11	10587 522 130	2022-06-02
master	v5.18-11972-gd1dc8776...	d1dc87763f406d4e67ca...	206 13 11	9425 429 121	2022-06-02
master	v5.18-11934-g54eb8462...	54eb8462f21fb170a05a...	206 13 11	6520 353 90	2022-06-02
master	v5.18-11429-ge11a9356...	e11a93567d3f1e843300...	200 13 11	13181 573 115	2022-06-01
master	v5.18-11439-g8ab2afa2...	8ab2afa23bd197df4781...	202 12 11	12937 587 115	2022-06-01



Home Jobs Builds Tests SoCs Info

Test Results: «v5.18-11817-g8171acb8bc9b3» (mainline / master)

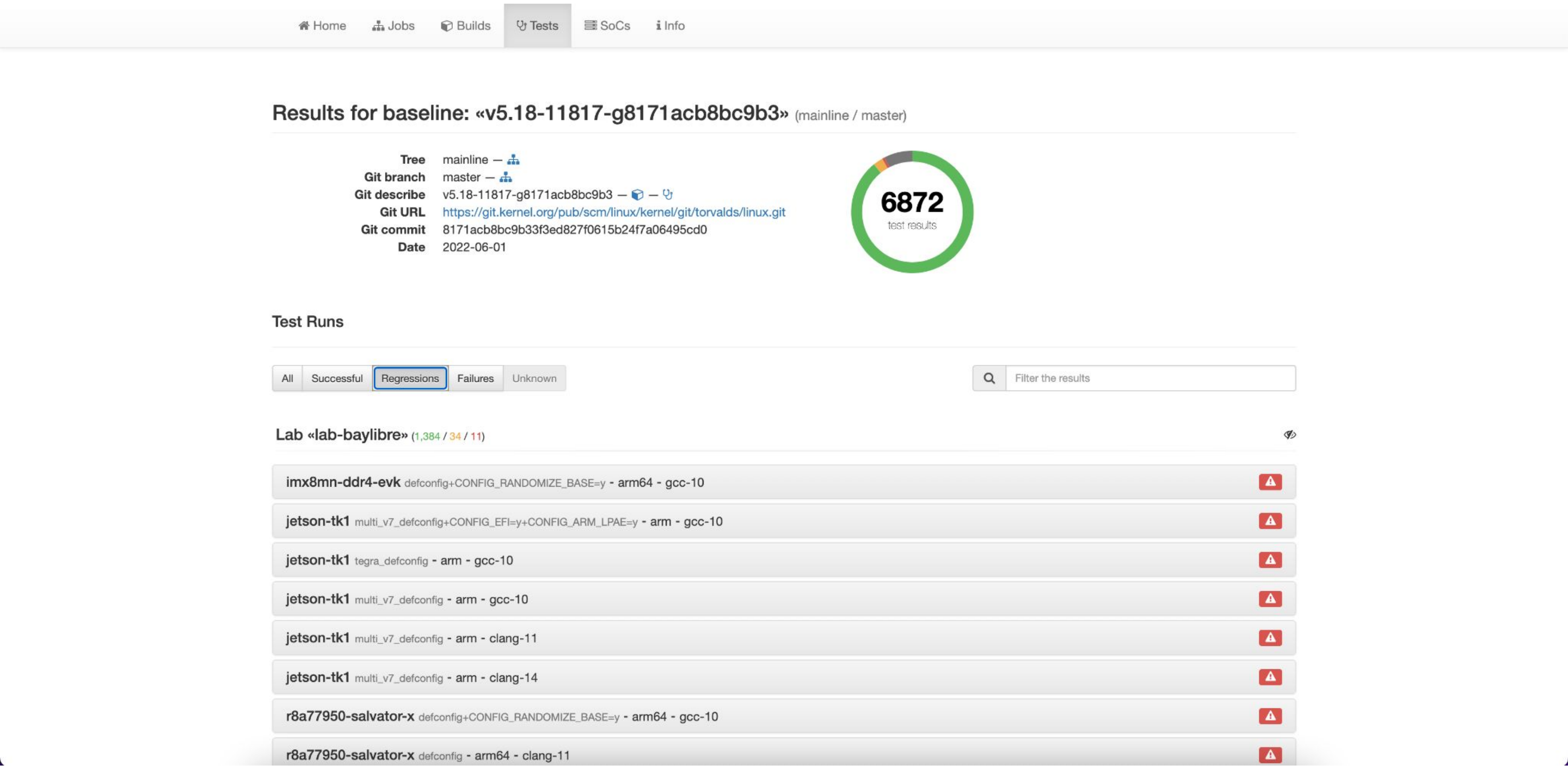
Tree	mainline —
Git branch	master —
Git describe	v5.18-11817-g8171acb8bc9b3 —
Git URL	https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git
Git commit	8171acb8bc9b33f3ed827f0615b24f7a06495cd0
Date	2022-06-01

12953
test results

Available Test Plans

Filter the results

Test Plan	Test Results	Status
baseline	6128 179 44	⚠
baseline-nfs	734 54 20	⚠
cros-ec	8 1 7	⚠
igt-gpu-amd	28 4 0	✖
igt-gpu-panfrost	15 1 4	⚠
igt-kms-exynos	132 3 0	✖
igt-kms-rockchip	75 14 3	⚠
igt-kms-tegra	0 0 3	⚠
kselftest-alsa	1531 41 2	⚠
kselftest-arm64	34 1 2	⚠
kselftest-cpufreq	4 0 2	⚠
kselftest-filesystems	16 6 1	⚠
kselftest-futex	34 5 4	⚠



Results for baseline: «v5.18-11817-g8171acb8bc9b3» (mainline / master)

Tree	mainline —
Git branch	master —
Git describe	v5.18-11817-g8171acb8bc9b3 — —
Git URL	https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git
Git commit	8171acb8bc9b33f3ed827f0615b24f7a06495cd0
Date	2022-06-01

Test Runs

Lab «lab-baylibre» (1,384 / 34 / 11)

imx8mn-ddr4-evk defconfig+CONFIG_RANDOMIZE_BASE=y - arm64 - gcc-10

jetson-tk1 multi_v7_defconfig+CONFIG_EFI=y+CONFIG_ARM_LPAE=y - arm - gcc-10

jetson-tk1 tegra_defconfig - arm - gcc-10

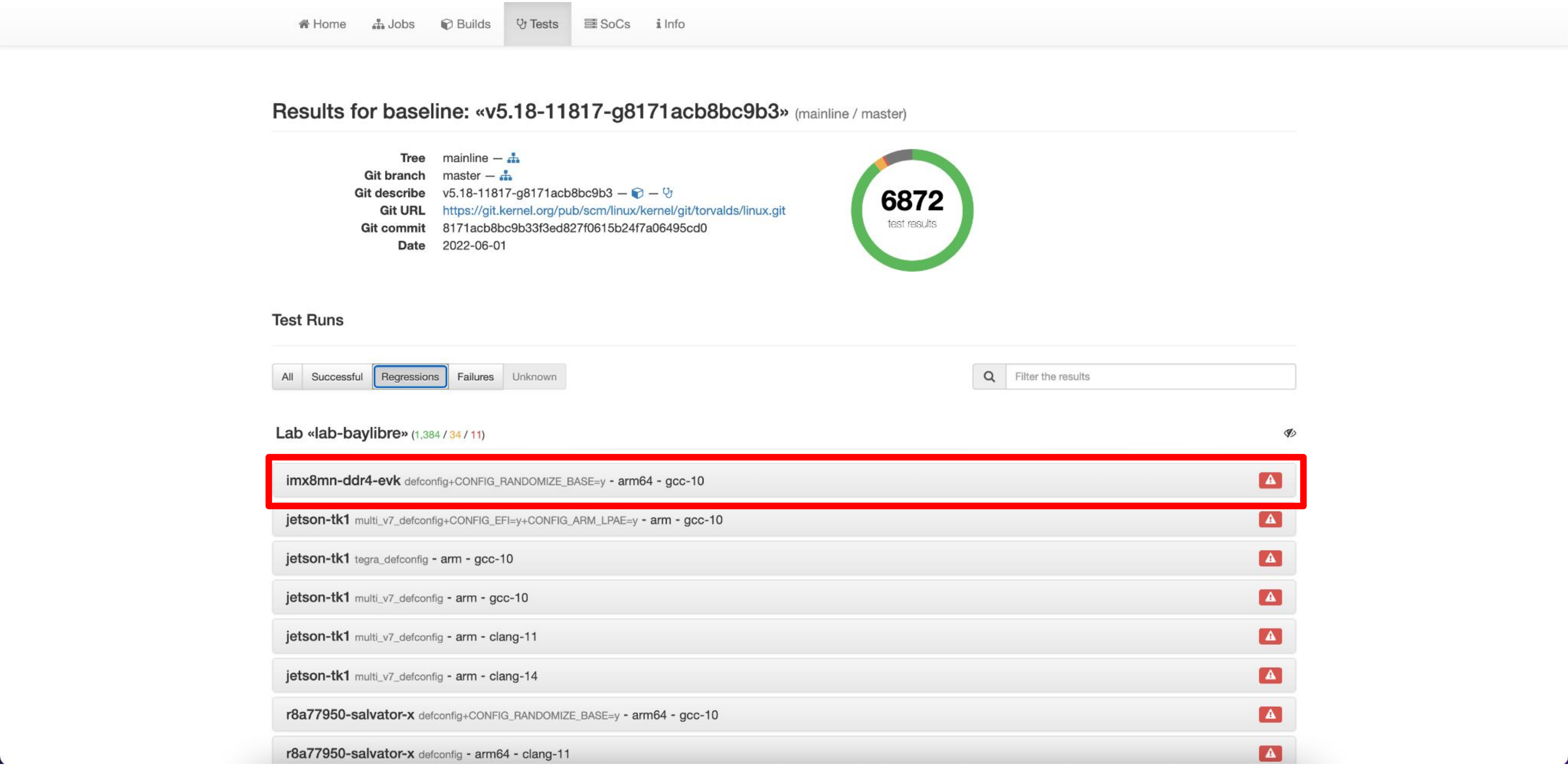
jetson-tk1 multi_v7_defconfig - arm - gcc-10

jetson-tk1 multi_v7_defconfig - arm - clang-11

jetson-tk1 multi_v7_defconfig - arm - clang-14

r8a77950-salvator-x defconfig+CONFIG_RANDOMIZE_BASE=y - arm64 - gcc-10

r8a77950-salvator-x defconfig - arm64 - clang-11



Home Jobs Builds Tests SoCs Info

Results for baseline: «v5.18-11817-g8171acb8bc9b3» (mainline / master)

Tree	mainline —
Git branch	master —
Git describe	v5.18-11817-g8171acb8bc9b3 — —
Git URL	https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git
Git commit	8171acb8bc9b33f3ed827f0615b24f7a06495cd0
Date	2022-06-01

6872
test results

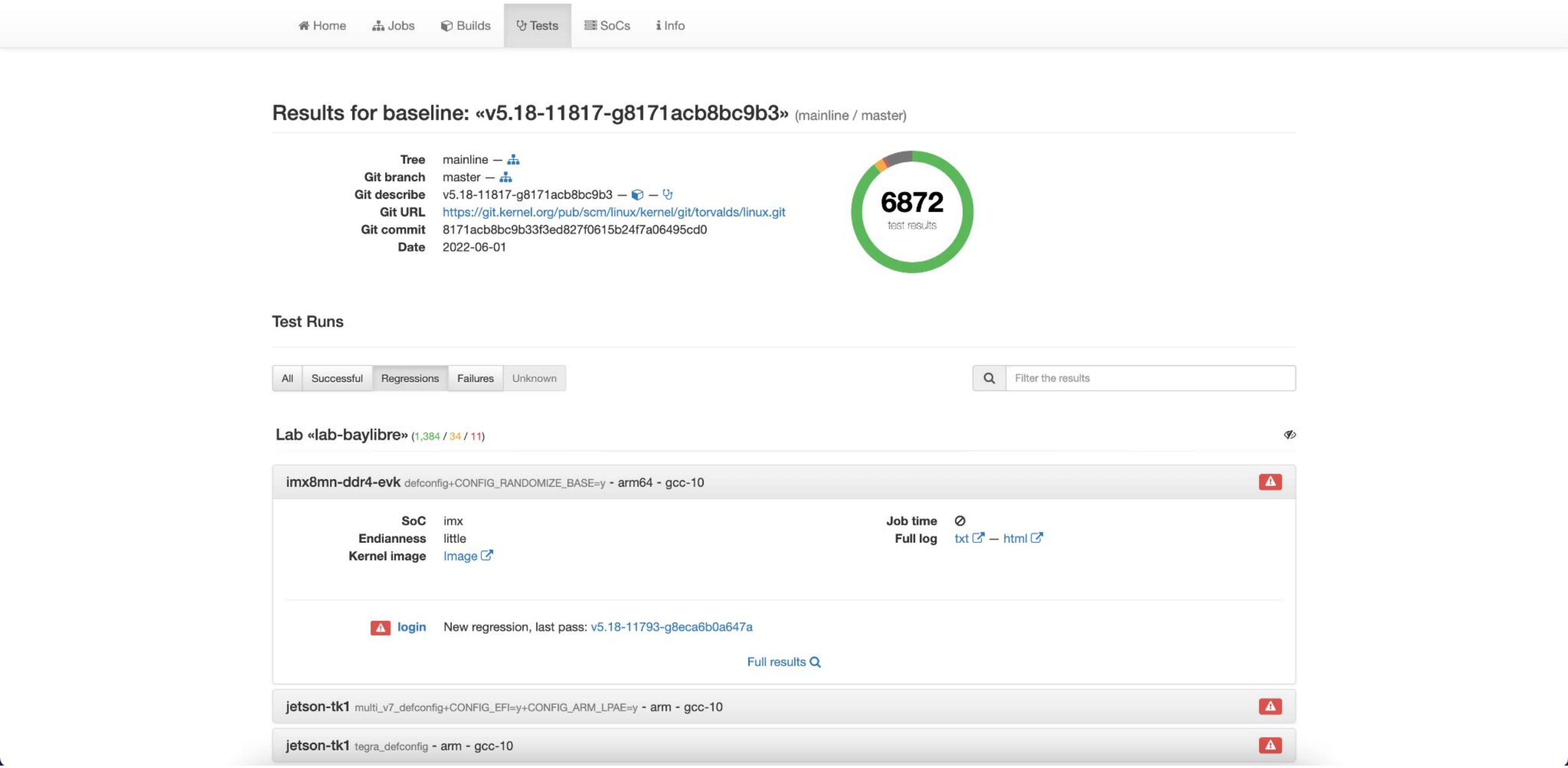
Test Runs

All | Successful | Regressions | Failures | Unknown

Filter the results

Lab «lab-baylibre» (1,384 / 34 / 11)

- **imx8mn-ddr4-evk** defconfig+CONFIG_RANDOMIZE_BASE=y - arm64 - gcc-10
- **jetson-tk1** multi_v7_defconfig+CONFIG_EFI=y+CONFIG_ARM_LPAE=y - arm - gcc-10
- **jetson-tk1** tegra_defconfig - arm - gcc-10
- **jetson-tk1** multi_v7_defconfig - arm - gcc-10
- **jetson-tk1** multi_v7_defconfig - arm - clang-11
- **jetson-tk1** multi_v7_defconfig - arm - clang-14
- **r8a77950-salvator-x** defconfig+CONFIG_RANDOMIZE_BASE=y - arm64 - gcc-10
- **r8a77950-salvator-x** defconfig - arm64 - clang-11



Home Jobs Builds Tests SoCs Info

Results for baseline: «v5.18-11817-g8171acb8bc9b3» (mainline / master)

Tree	mainline —
Git branch	master —
Git describe	v5.18-11817-g8171acb8bc9b3 — —
Git URL	https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git
Git commit	8171acb8bc9b33f3ed827f0615b24f7a06495cd0
Date	2022-06-01

Test Runs

Lab «lab-baylibre» (1,384 / 34 / 11)

imx8mn-ddr4-evk defconfig+CONFIG_RANDOMIZE_BASE=y - arm64 - gcc-10

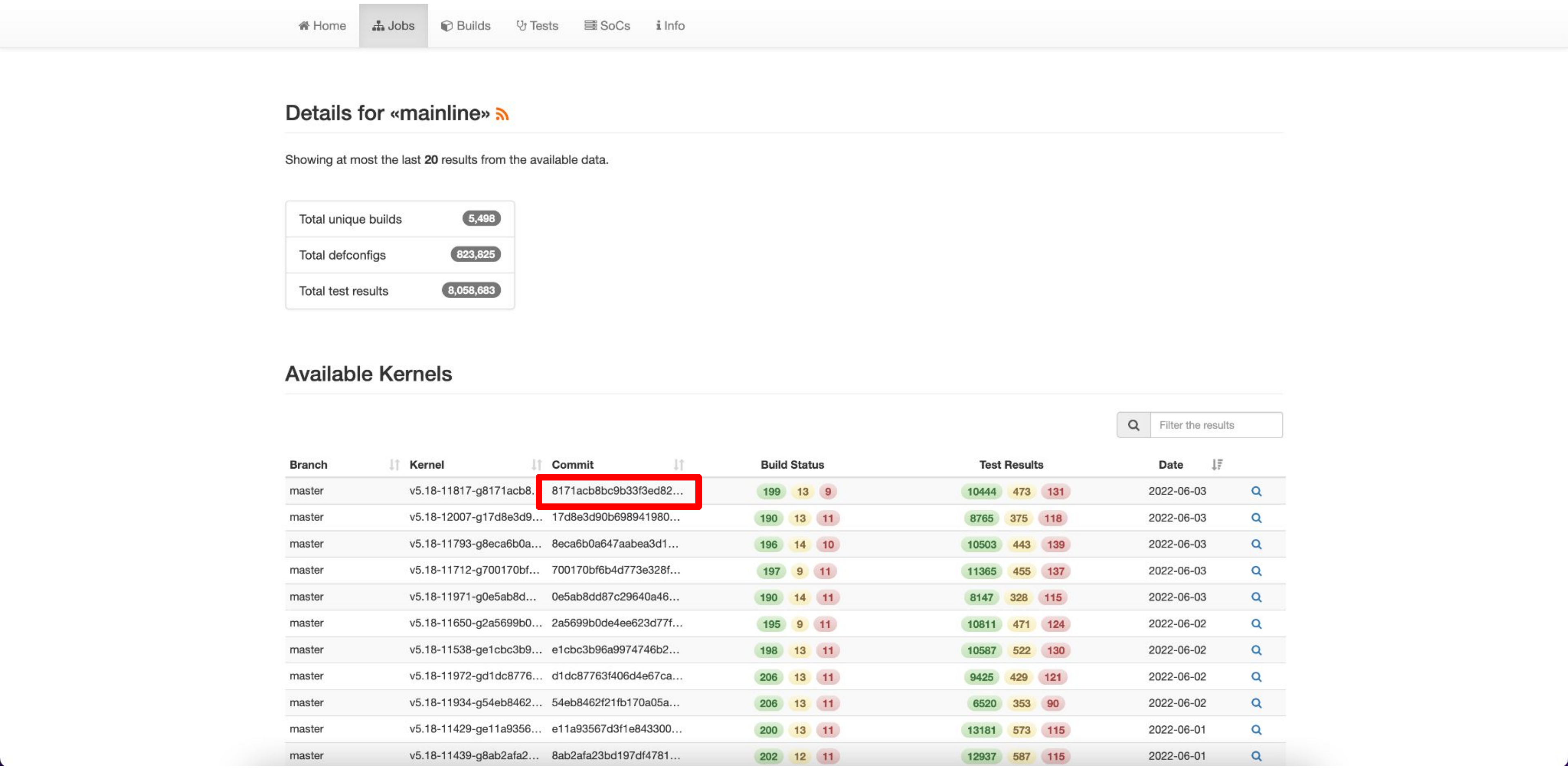
SoC	imx	**Job time**	⊘
Endianness	little	**Full log**	txt — html
Kernel image	Image		

login New regression, last pass: v5.18-11793-g8eca6b0a647a

Full results

jetson-tk1 multi_v7_defconfig+CONFIG_EFI=y+CONFIG_ARM_LPAE=y - arm - gcc-10

jetson-tk1 tegra_defconfig - arm - gcc-10



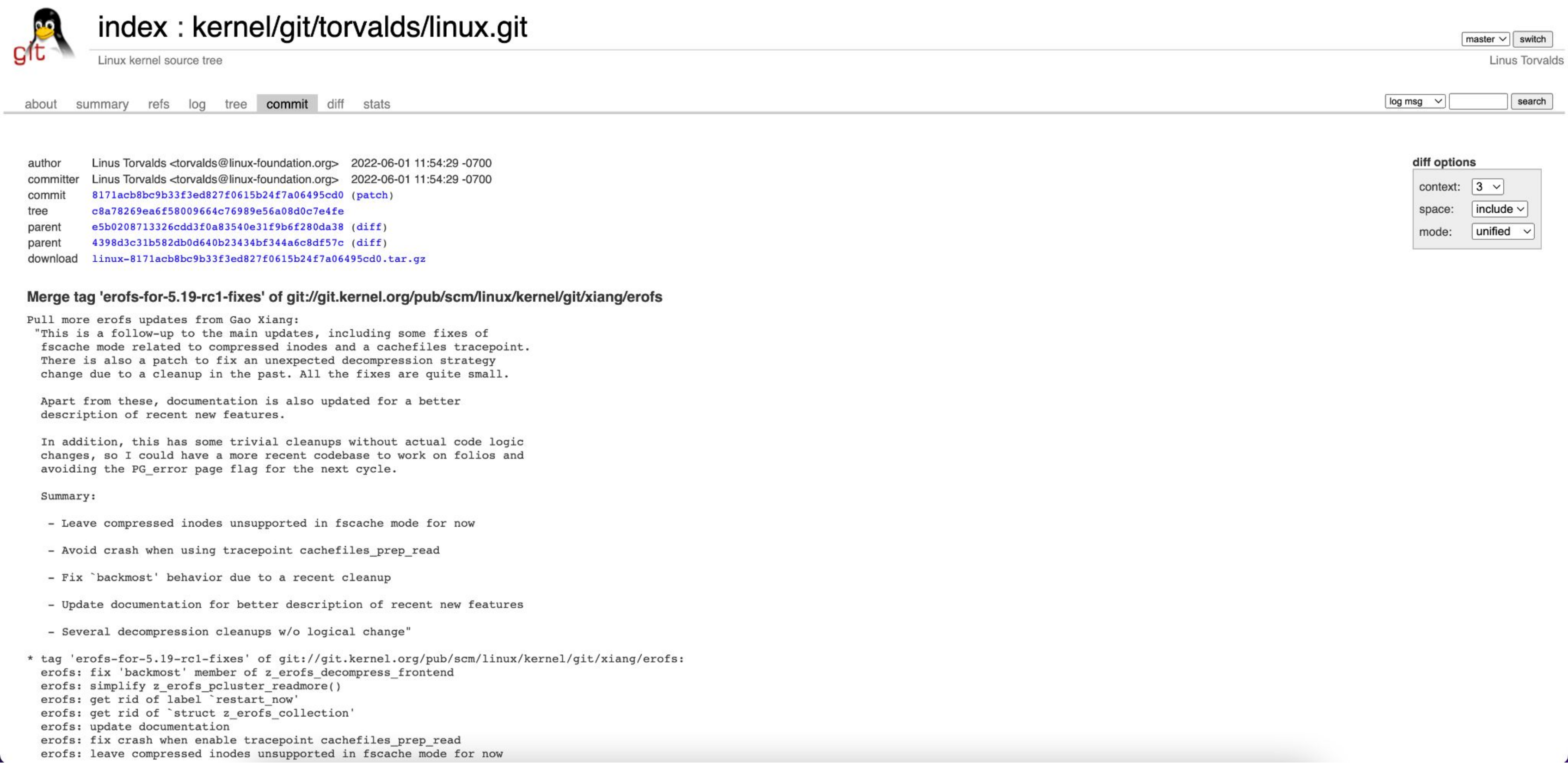
Details for «mainline»

Showing at most the last **20** results from the available data.

Total unique builds	5,498
Total defconfigs	823,825
Total test results	8,058,683

Available Kernels

Branch	Kernel	Commit	Build Status			Test Results			Date
master	v5.18-11817-g8171acb8...	8171acb8bc9b33f3ed82...	199	13	9	10444	473	131	2022-06-03
master	v5.18-12007-g17d8e3d9...	17d8e3d90b698941980...	190	13	11	8765	375	118	2022-06-03
master	v5.18-11793-g8eca6b0a...	8eca6b0a647aabea3d1...	196	14	10	10503	443	139	2022-06-03
master	v5.18-11712-g700170bf...	700170bf6b4d773e328f...	197	9	11	11365	455	137	2022-06-03
master	v5.18-11971-g0e5ab8d...	0e5ab8dd87c29640a46...	190	14	11	8147	328	115	2022-06-03
master	v5.18-11650-g2a5699b0...	2a5699b0de4ee623d77f...	195	9	11	10811	471	124	2022-06-02
master	v5.18-11538-ge1cbc3b9...	e1cbc3b96a9974746b2...	198	13	11	10587	522	130	2022-06-02
master	v5.18-11972-gd1dc8776...	d1dc87763f406d4e67ca...	206	13	11	9425	429	121	2022-06-02
master	v5.18-11934-g54eb8462...	54eb8462f21fb170a05a...	206	13	11	6520	353	90	2022-06-02
master	v5.18-11429-ge11a9356...	e11a93567d3f1e843300...	200	13	11	13181	573	115	2022-06-01
master	v5.18-11439-g8ab2afa2...	8ab2afa23bd197df4781...	202	12	11	12937	587	115	2022-06-01

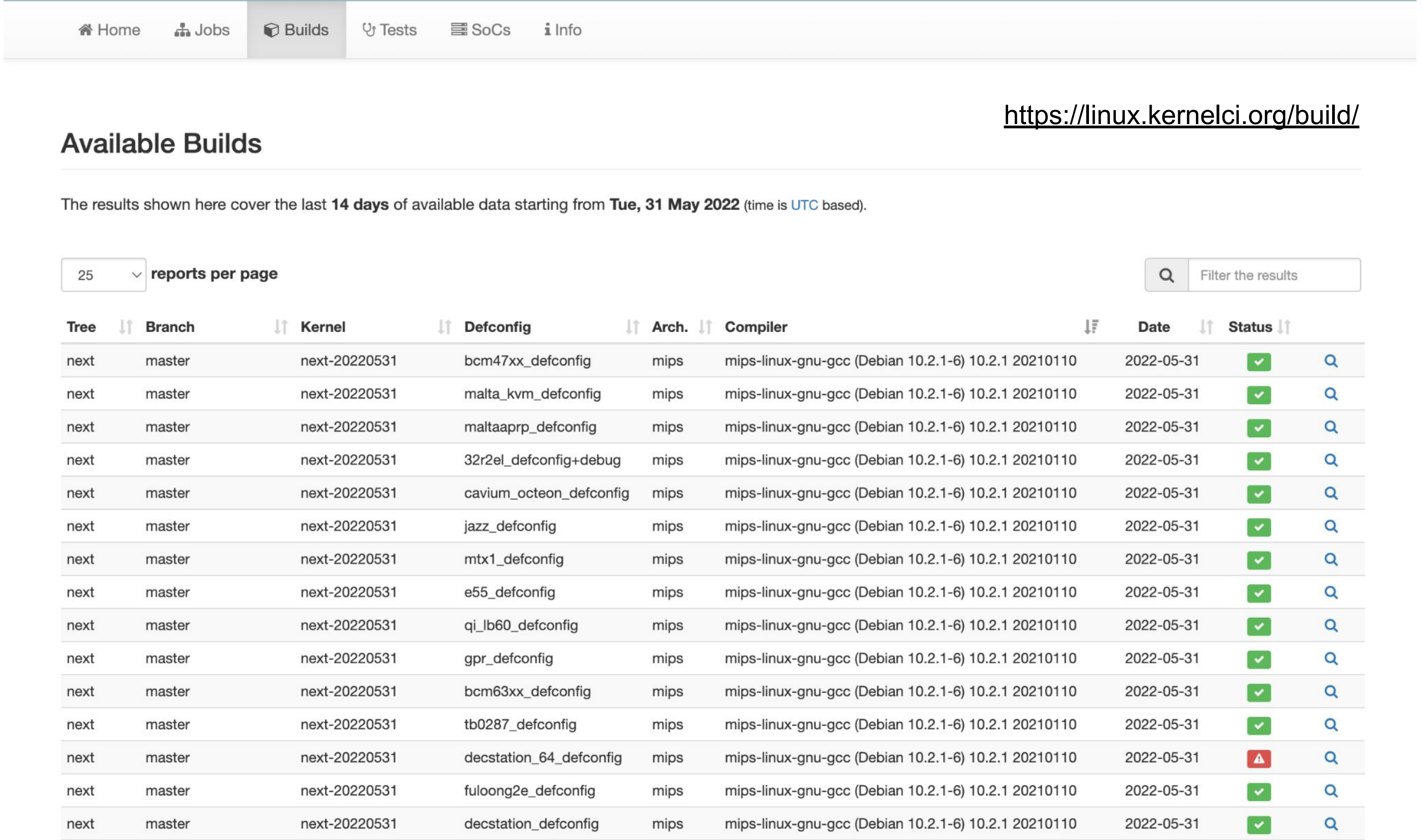


author Linus Torvalds <torvalds@linux-foundation.org> 2022-06-01 11:54:29 -0700
committer Linus Torvalds <torvalds@linux-foundation.org> 2022-06-01 11:54:29 -0700
commit 8171acb8bc9b33f3ed827f0615b24f7a06495cd0 (patch)
tree c8a78269ea6f58009664c76989e56a08d0c7e4fe
parent e5b0208713326cdd3f0a83540e31f9b6f280da38 (diff)
parent 4398d3c31b582db0d640b23434bf344a6c8df57c (diff)
download linux-8171acb8bc9b33f3ed827f0615b24f7a06495cd0.tar.gz

diff options
context: 3
space: include
mode: unified

Merge tag 'erofs-for-5.19-rc1-fixes' of git://git.kernel.org/pub/scm/linux/kernel/git/xiang/erofs

```
Pull more erofs updates from Gao Xiang:
 "This is a follow-up to the main updates, including some fixes of
  fscache mode related to compressed inodes and a cachefiles tracepoint.
  There is also a patch to fix an unexpected decompression strategy
  change due to a cleanup in the past. All the fixes are quite small.

  Apart from these, documentation is also updated for a better
  description of recent new features.

  In addition, this has some trivial cleanups without actual code logic
  changes, so I could have a more recent codebase to work on folios and
  avoiding the PG_error page flag for the next cycle.

  Summary:

   - Leave compressed inodes unsupported in fscache mode for now

   - Avoid crash when using tracepoint cachefiles_prep_read

   - Fix `backmost' behavior due to a recent cleanup

   - Update documentation for better description of recent new features

   - Several decompression cleanups w/o logical change"

* tag 'erofs-for-5.19-rc1-fixes' of git://git.kernel.org/pub/scm/linux/kernel/git/xiang/erofs:
  erofs: fix 'backmost' member of z_erofs_decompress_frontend
  erofs: simplify z_erofs_pcluster_readmore()
  erofs: get rid of label `restart_now'
  erofs: get rid of `struct z_erofs_collection'
  erofs: update documentation
  erofs: fix crash when enable tracepoint cachefiles_prep_read
  erofs: leave compressed inodes unsupported in fscache mode for now
```



https://linux.kernelci.org/build/

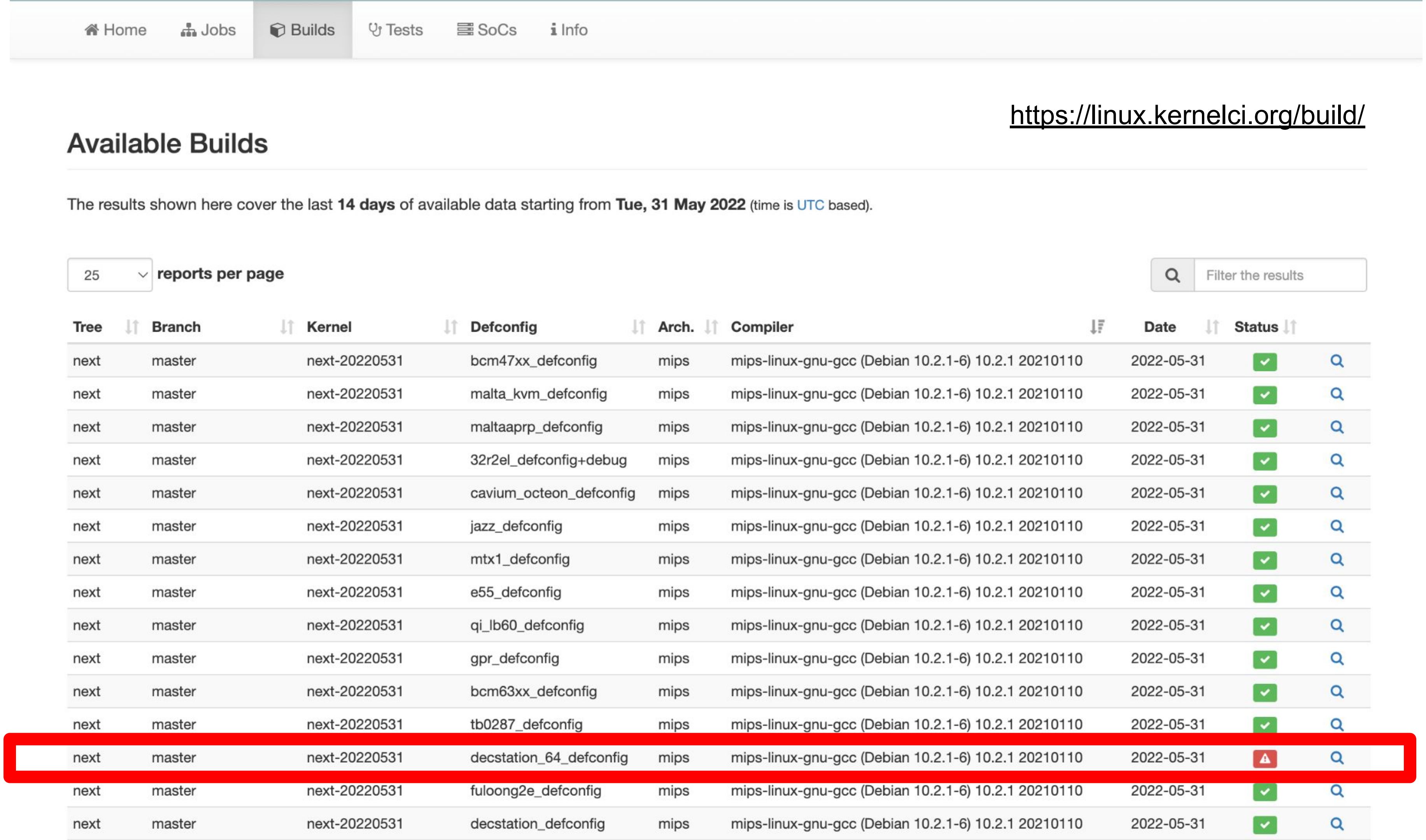
Available Builds

The results shown here cover the last **14 days** of available data starting from **Tue, 31 May 2022** (time is UTC based).

25 **reports per page**

Filter the results

Tree	Branch	Kernel	Defconfig	Arch.	Compiler	Date	Status	
next	master	next-20220531	bcm47xx_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	malta_kvm_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	maltaaprp_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	32r2el_defconfig+debug	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	cavium_octeon_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	jazz_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	mtx1_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	e55_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	qi_lb60_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	gpr_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	bcm63xx_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	tb0287_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	decstation_64_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	⚠	🔍
next	master	next-20220531	fuloong2e_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	decstation_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍



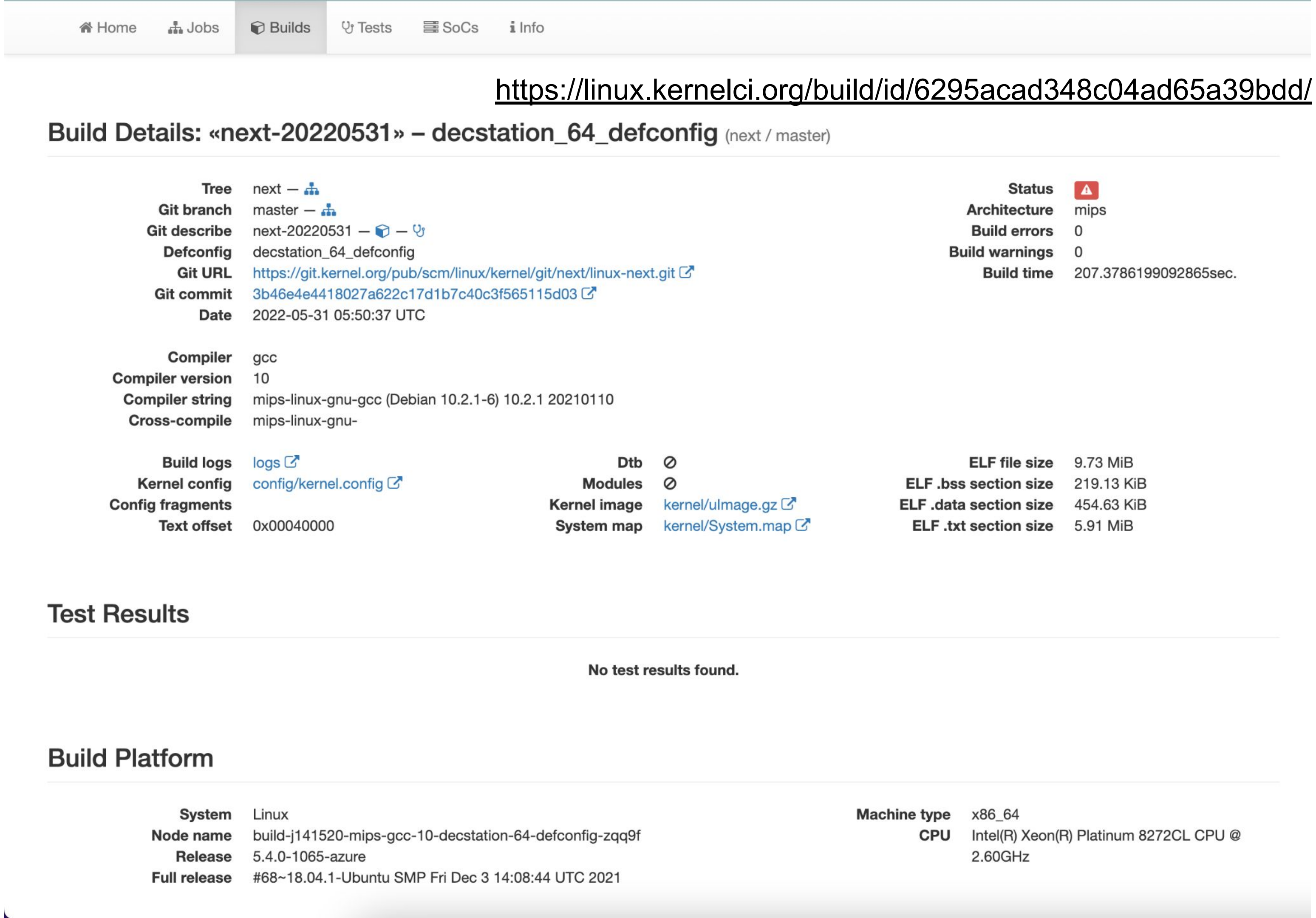
https://linux.kernelci.org/build/

Available Builds

The results shown here cover the last **14 days** of available data starting from **Tue, 31 May 2022** (time is UTC based).

Tree	Branch	Kernel	Defconfig	Arch.	Compiler	Date	Status	
next	master	next-20220531	bcm47xx_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	malta_kvm_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	maltaaprp_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	32r2el_defconfig+debug	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	cavium_octeon_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	jazz_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	mtx1_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	e55_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	qi_lb60_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	gpr_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	bcm63xx_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	tb0287_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	decstation_64_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	⚠	🔍
next	master	next-20220531	fuloong2e_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍
next	master	next-20220531	decstation_defconfig	mips	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110	2022-05-31	✓	🔍

https://linux.kernelci.org/build/id/6295acad348c04ad65a39bdd/



Build Details: «next-20220531» – decstation_64_defconfig (next / master)

Tree	next —
Git branch	master —
Git describe	next-20220531 — —
Defconfig	decstation_64_defconfig
Git URL	https://git.kernel.org/pub/scm/linux/kernel/git/next/linux-next.git
Git commit	3b46e4e4418027a622c17d1b7c40c3f565115d03
Date	2022-05-31 05:50:37 UTC
Compiler	gcc
Compiler version	10
Compiler string	mips-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110
Cross-compile	mips-linux-gnu-

Status	⚠
Architecture	mips
Build errors	0
Build warnings	0
Build time	207.3786199092865sec.

Build logs	logs	**Dtb**	⊘	**ELF file size**	9.73 MiB
Kernel config	config/kernel.config	**Modules**	⊘	**ELF .bss section size**	219.13 KiB
Config fragments		**Kernel image**	kernel/uImage.gz	**ELF .data section size**	454.63 KiB
Text offset	0x00040000	**System map**	kernel/System.map	**ELF .txt section size**	5.91 MiB

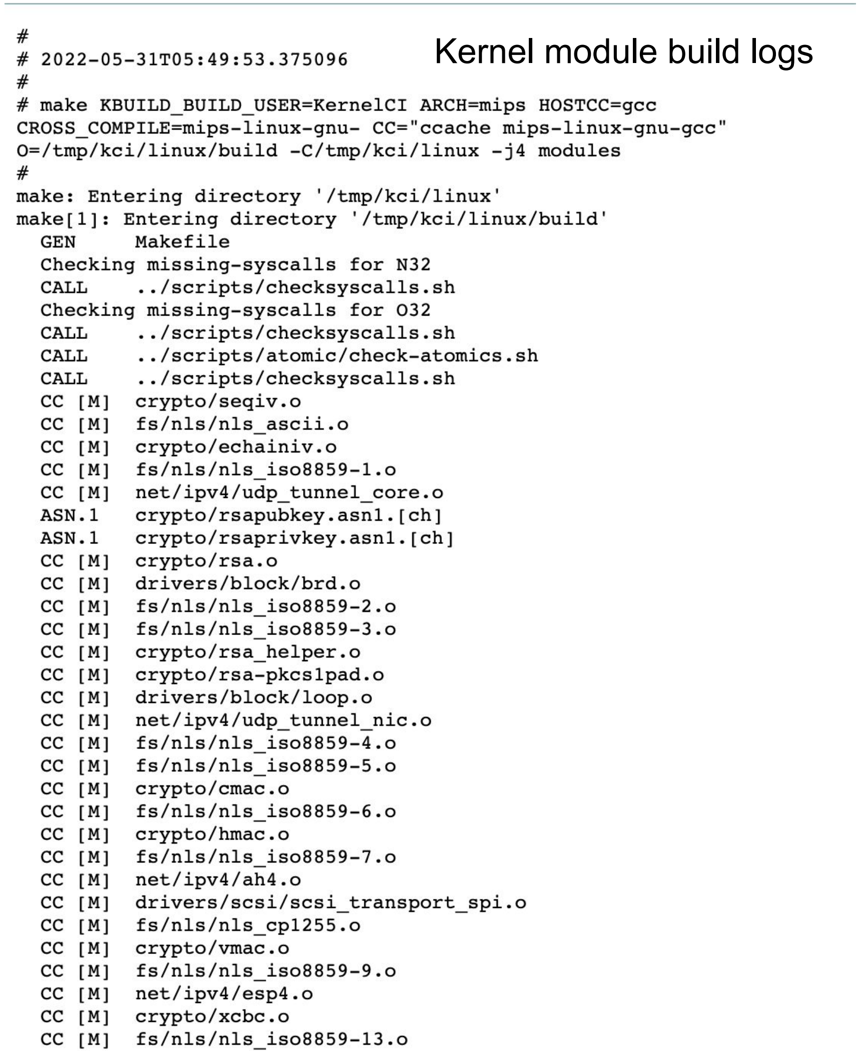
Test Results

No test results found.

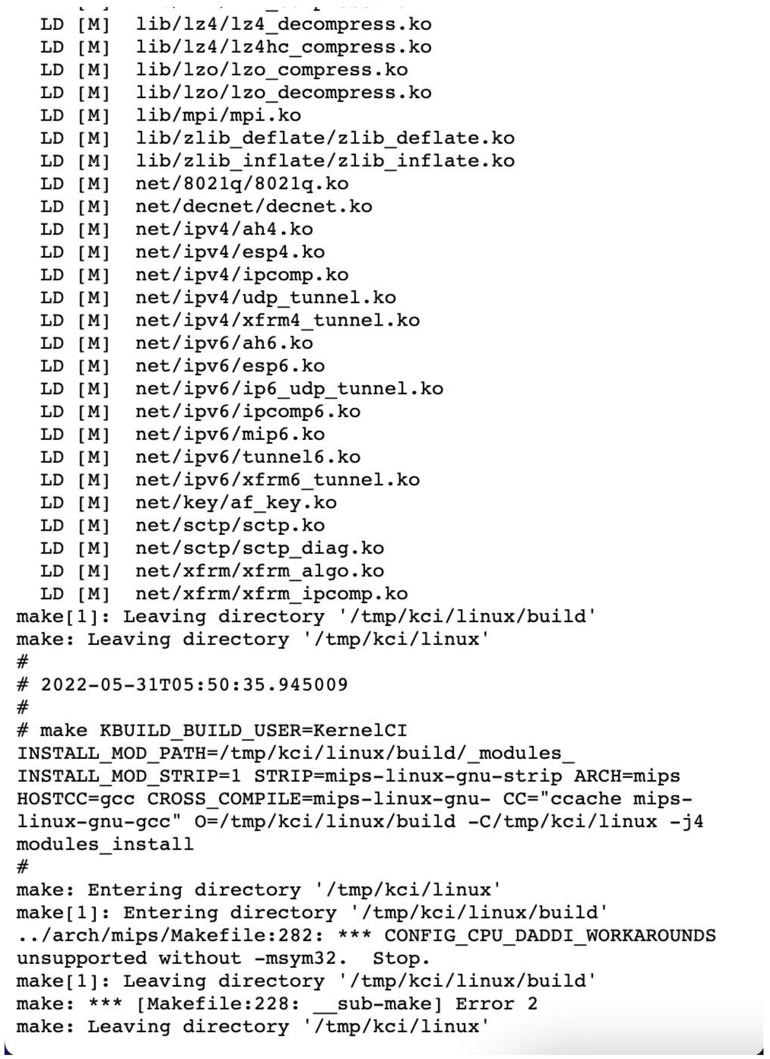
Build Platform

System	Linux	**Machine type**	x86_64
Node name	build-j141520-mips-gcc-10-decstation-64-defconfig-zqq9f	**CPU**	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz
Release	5.4.0-1065-azure		
Full release	#68~18.04.1-Ubuntu SMP Fri Dec 3 14:08:44 UTC 2021		

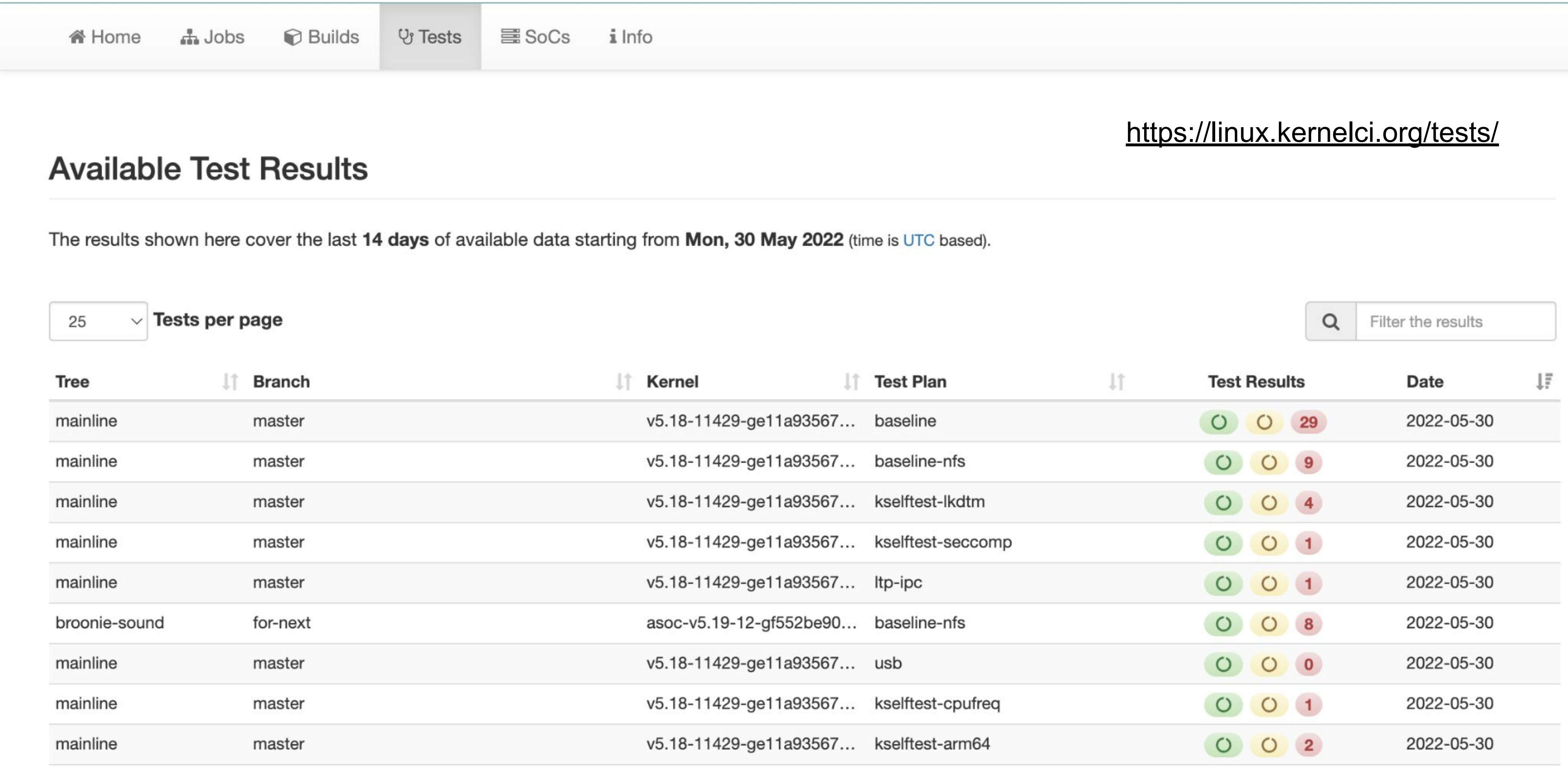
Kernel module build logs



```
#
# 2022-05-31T05:49:53.375096
#
# make KBUILD_BUILD_USER=KernelCI ARCH=mips HOSTCC=gcc
CROSS_COMPILE=mips-linux-gnu- CC="ccache mips-linux-gnu-gcc"
O=/tmp/kci/linux/build -C/tmp/kci/linux -j4 modules
#
make: Entering directory '/tmp/kci/linux'
make[1]: Entering directory '/tmp/kci/linux/build'
  GEN     Makefile
  Checking missing-syscalls for N32
  CALL    ../scripts/checksyscalls.sh
  Checking missing-syscalls for O32
  CALL    ../scripts/checksyscalls.sh
  CALL    ../scripts/atomic/check-atomics.sh
  CALL    ../scripts/checksyscalls.sh
  CC [M]  crypto/seqiv.o
  CC [M]  fs/nls/nls_ascii.o
  CC [M]  crypto/echainiv.o
  CC [M]  fs/nls/nls_iso8859-1.o
  CC [M]  net/ipv4/udp_tunnel_core.o
  ASN.1   crypto/rsapubkey.asn1.[ch]
  ASN.1   crypto/rsaprivkey.asn1.[ch]
  CC [M]  crypto/rsa.o
  CC [M]  drivers/block/brd.o
  CC [M]  fs/nls/nls_iso8859-2.o
  CC [M]  fs/nls/nls_iso8859-3.o
  CC [M]  crypto/rsa_helper.o
  CC [M]  crypto/rsa-pkcs1pad.o
  CC [M]  drivers/block/loop.o
  CC [M]  net/ipv4/udp_tunnel_nic.o
  CC [M]  fs/nls/nls_iso8859-4.o
  CC [M]  fs/nls/nls_iso8859-5.o
  CC [M]  crypto/cmac.o
  CC [M]  fs/nls/nls_iso8859-6.o
  CC [M]  crypto/hmac.o
  CC [M]  fs/nls/nls_iso8859-7.o
  CC [M]  net/ipv4/ah4.o
  CC [M]  drivers/scsi/scsi_transport_spi.o
  CC [M]  fs/nls/nls_cp1255.o
  CC [M]  crypto/vmac.o
  CC [M]  fs/nls/nls_iso8859-9.o
  CC [M]  net/ipv4/esp4.o
  CC [M]  crypto/xcbc.o
  CC [M]  fs/nls/nls_iso8859-13.o
```



```
  LD [M]  lib/lz4/lz4_decompress.ko
  LD [M]  lib/lz4/lz4hc_compress.ko
  LD [M]  lib/lzo/lzo_compress.ko
  LD [M]  lib/lzo/lzo_decompress.ko
  LD [M]  lib/mpi/mpi.ko
  LD [M]  lib/zlib_deflate/zlib_deflate.ko
  LD [M]  lib/zlib_inflate/zlib_inflate.ko
  LD [M]  net/8021q/8021q.ko
  LD [M]  net/decnet/decnet.ko
  LD [M]  net/ipv4/ah4.ko
  LD [M]  net/ipv4/esp4.ko
  LD [M]  net/ipv4/ipcomp.ko
  LD [M]  net/ipv4/udp_tunnel.ko
  LD [M]  net/ipv4/xfrm4_tunnel.ko
  LD [M]  net/ipv6/ah6.ko
  LD [M]  net/ipv6/esp6.ko
  LD [M]  net/ipv6/ip6_udp_tunnel.ko
  LD [M]  net/ipv6/ipcomp6.ko
  LD [M]  net/ipv6/mip6.ko
  LD [M]  net/ipv6/tunnel6.ko
  LD [M]  net/ipv6/xfrm6_tunnel.ko
  LD [M]  net/key/af_key.ko
  LD [M]  net/sctp/sctp.ko
  LD [M]  net/sctp/sctp_diag.ko
  LD [M]  net/xfrm/xfrm_algo.ko
  LD [M]  net/xfrm/xfrm_ipcomp.ko
make[1]: Leaving directory '/tmp/kci/linux/build'
make: Leaving directory '/tmp/kci/linux'
#
# 2022-05-31T05:50:35.945009
#
# make KBUILD_BUILD_USER=KernelCI
INSTALL_MOD_PATH=/tmp/kci/linux/build/_modules_
INSTALL_MOD_STRIP=1 STRIP=mips-linux-gnu-strip ARCH=mips
HOSTCC=gcc CROSS_COMPILE=mips-linux-gnu- CC="ccache mips-
linux-gnu-gcc" O=/tmp/kci/linux/build -C/tmp/kci/linux -j4
modules_install
#
make: Entering directory '/tmp/kci/linux'
make[1]: Entering directory '/tmp/kci/linux/build'
../arch/mips/Makefile:282: *** CONFIG_CPU_DADDI_WORKAROUNDS
unsupported without -msym32.  Stop.
make[1]: Leaving directory '/tmp/kci/linux/build'
make: *** [Makefile:228: __sub-make] Error 2
make: Leaving directory '/tmp/kci/linux'
```



Home Jobs Builds Tests SoCs Info

https://linux.kernelci.org/tests/

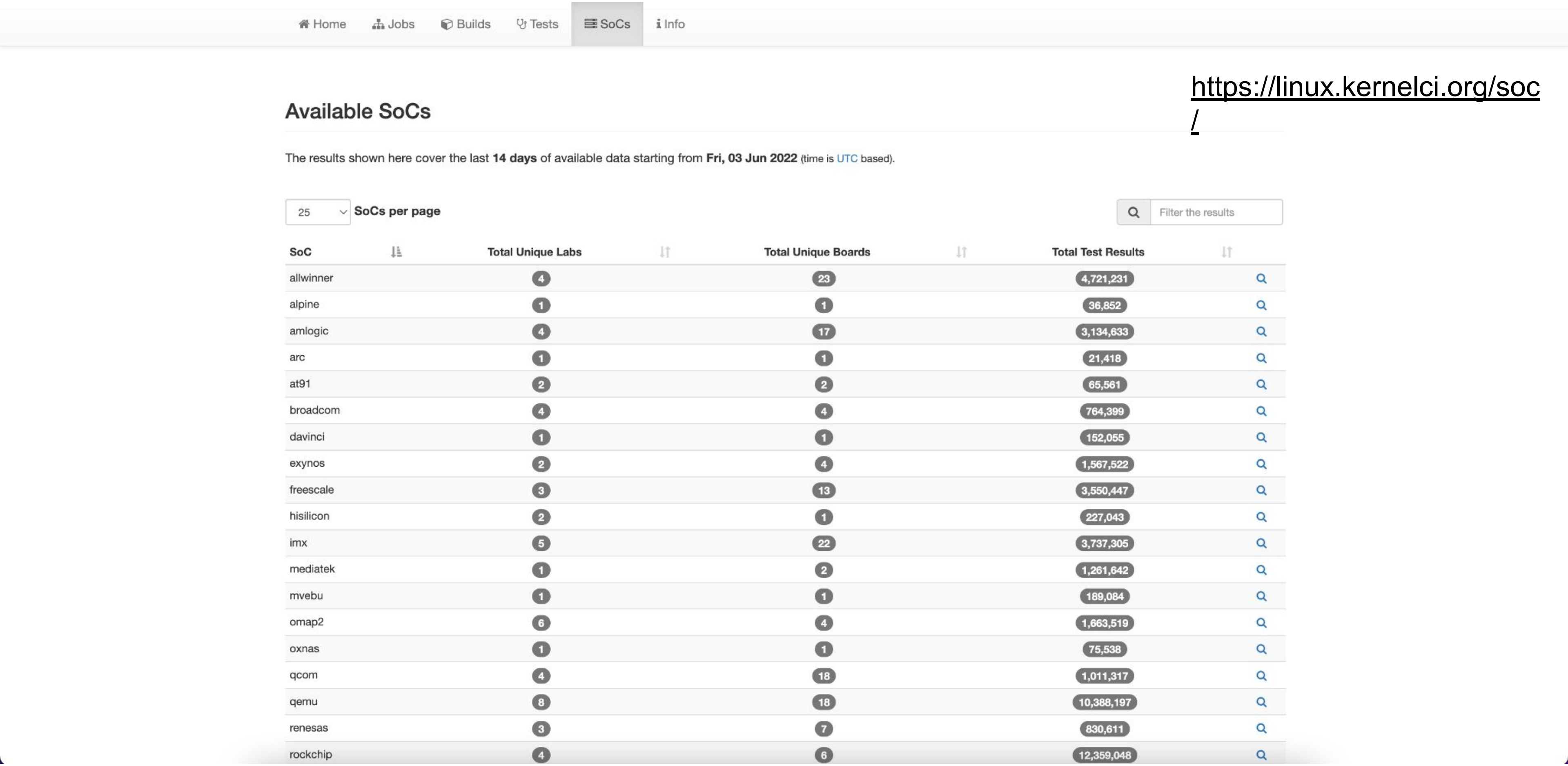
Available Test Results

The results shown here cover the last **14 days** of available data starting from **Mon, 30 May 2022** (time is UTC based).

25 **Tests per page**

Filter the results

Tree	Branch	Kernel	Test Plan	Test Results	Date
mainline	master	v5.18-11429-ge11a93567…	baseline	○ ○ 29	2022-05-30
mainline	master	v5.18-11429-ge11a93567…	baseline-nfs	○ ○ 9	2022-05-30
mainline	master	v5.18-11429-ge11a93567…	kselftest-lkdtm	○ ○ 4	2022-05-30
mainline	master	v5.18-11429-ge11a93567…	kselftest-seccomp	○ ○ 1	2022-05-30
mainline	master	v5.18-11429-ge11a93567…	ltp-ipc	○ ○ 1	2022-05-30
broonie-sound	for-next	asoc-v5.19-12-gf552be90…	baseline-nfs	○ ○ 8	2022-05-30
mainline	master	v5.18-11429-ge11a93567…	usb	○ ○ 0	2022-05-30
mainline	master	v5.18-11429-ge11a93567…	kselftest-cpufreq	○ ○ 1	2022-05-30
mainline	master	v5.18-11429-ge11a93567…	kselftest-arm64	○ ○ 2	2022-05-30


Home Jobs Builds Tests SoCs Info

https://linux.kernelci.org/soc/

Available SoCs

The results shown here cover the last **14 days** of available data starting from **Fri, 03 Jun 2022** (time is UTC based).

25 **SoCs per page**

Filter the results

SoC	Total Unique Labs	Total Unique Boards	Total Test Results
allwinner	4	23	4,721,231
alpine	1	1	36,852
amlogic	4	17	3,134,633
arc	1	1	21,418
at91	2	2	65,561
broadcom	4	4	764,399
davinci	1	1	152,055
exynos	2	4	1,567,522
freescale	3	13	3,550,447
hisilicon	2	1	227,043
imx	5	22	3,737,305
mediatek	1	2	1,261,642
mvebu	1	1	189,084
omap2	6	4	1,663,519
oxnas	1	1	75,538
qcom	4	18	1,011,317
qemu	8	18	10,388,197
renesas	3	7	830,611
rockchip	4	6	12,359,048

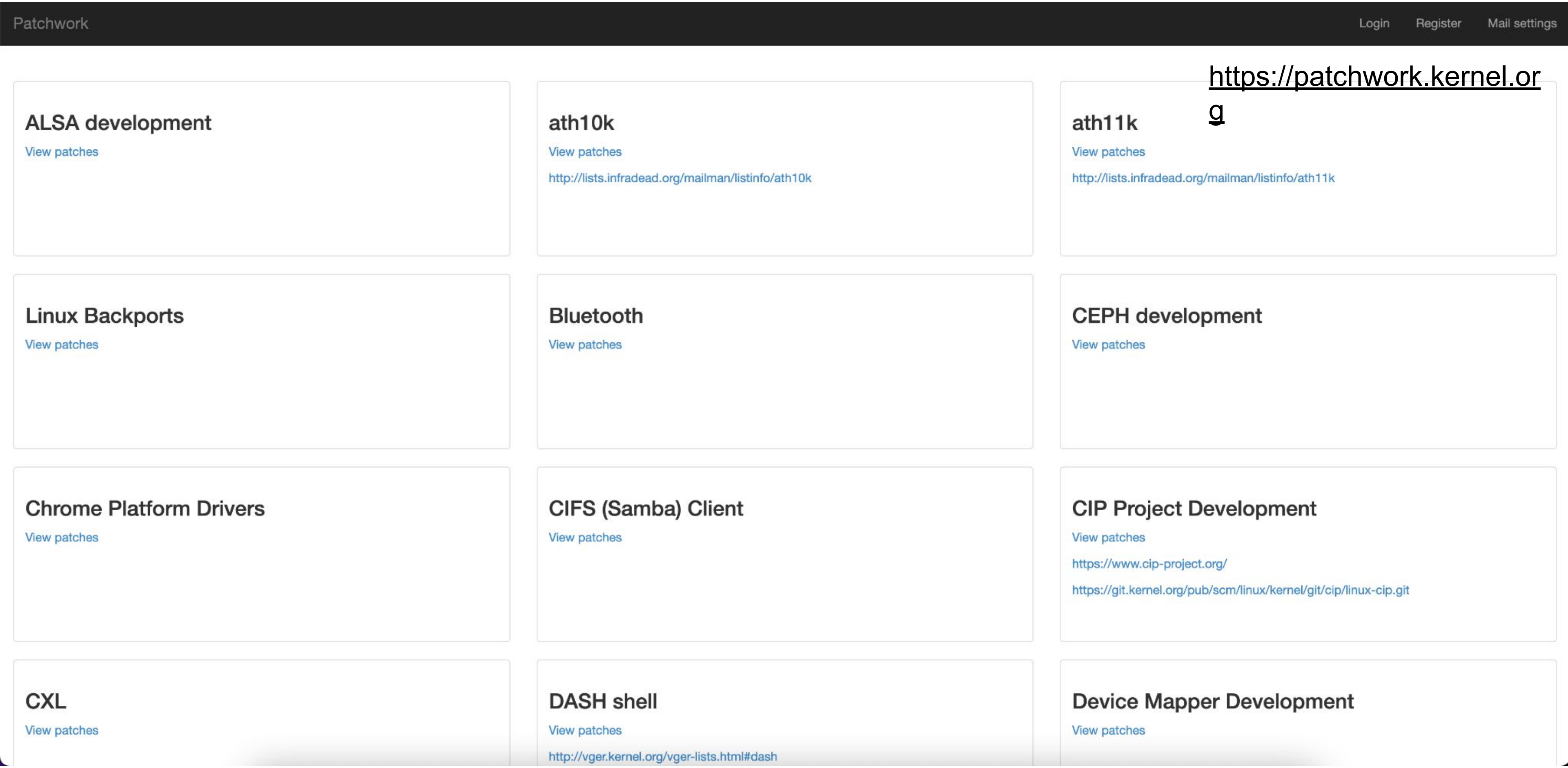
KernelCI – Pros and Cons

Pros

- Builds for multiple architectures
- Tests on multiple architectures
- Builds with multiple toolchains
- Useful information provided with failures and known regressions
- Open source and part of the Linux Foundation
- Emails failures to upstream lists
- Bisects to find culprit patches

Cons

- Only runs on merged patches
  - …but new APIs are coming to allow developers to address this
- Web dashboard needs some redesign, still has some bugs



Patchwork

Login Register Mail settings

https://patchwork.kernel.org

ALSA development
View patches

ath10k
View patches
http://lists.infradead.org/mailman/listinfo/ath10k

ath11k
View patches
http://lists.infradead.org/mailman/listinfo/ath11k

Linux Backports
View patches

Bluetooth
View patches

CEPH development
View patches

Chrome Platform Drivers
View patches

CIFS (Samba) Client
View patches

CIP Project Development
View patches
https://www.cip-project.org/
https://git.kernel.org/pub/scm/linux/kernel/git/cip/linux-cip.git

CXL
View patches

DASH shell
View patches
http://vger.kernel.org/vger-lists.html#dash

Device Mapper Development
View patches

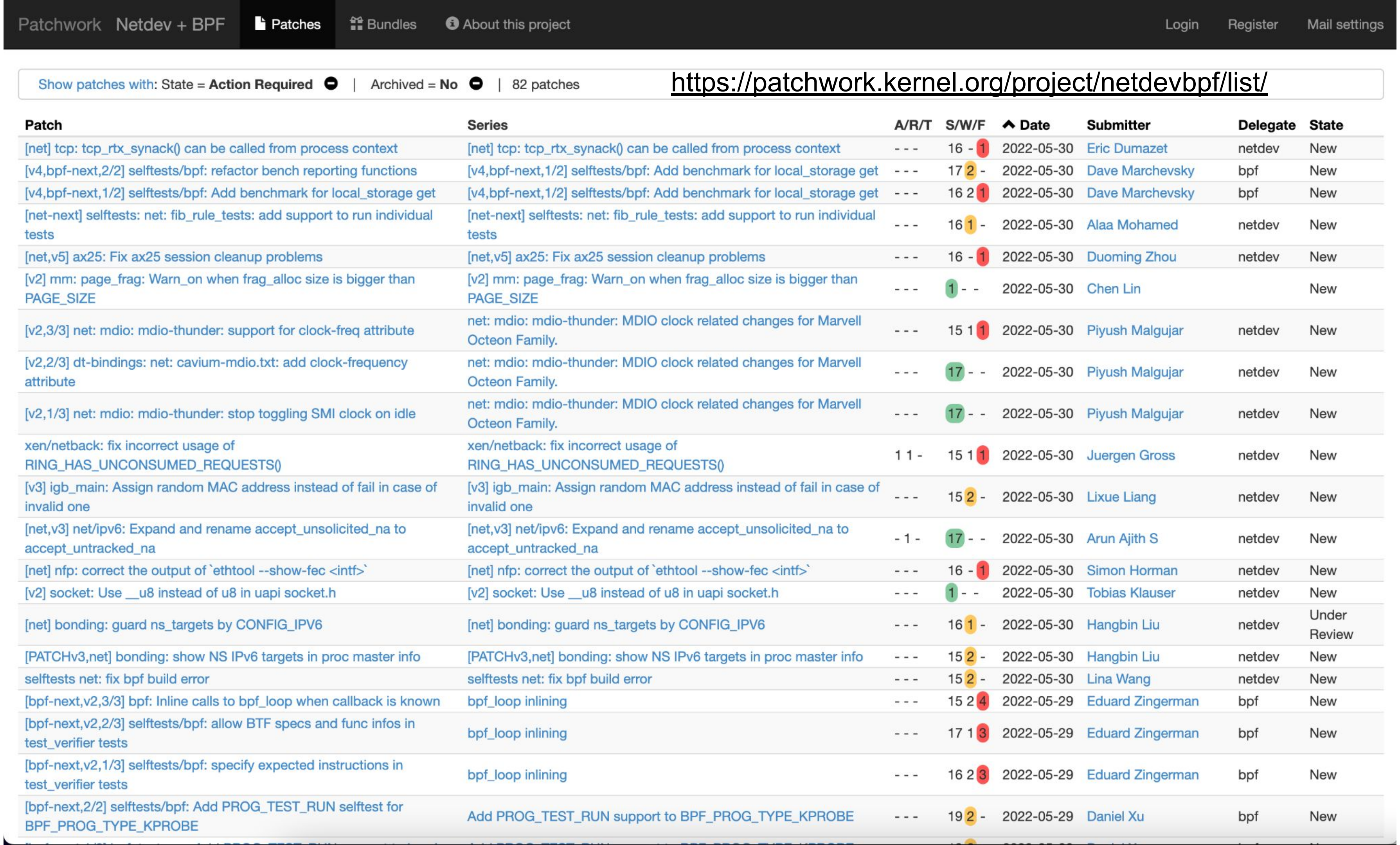
Patchwork + github – How BPF runs CI tests

Patchwork is a free, web-based patch tracking system

Architecture is a combination of patchwork, github, Meta infrastructure

Runs all BPF seltests (https://github.com/torvalds/linux/tree/master/tools/testing/selftests/bpf) on every patch sent to bpf and bpf-next lists

Only builds and tests for x86 and s390x architectures

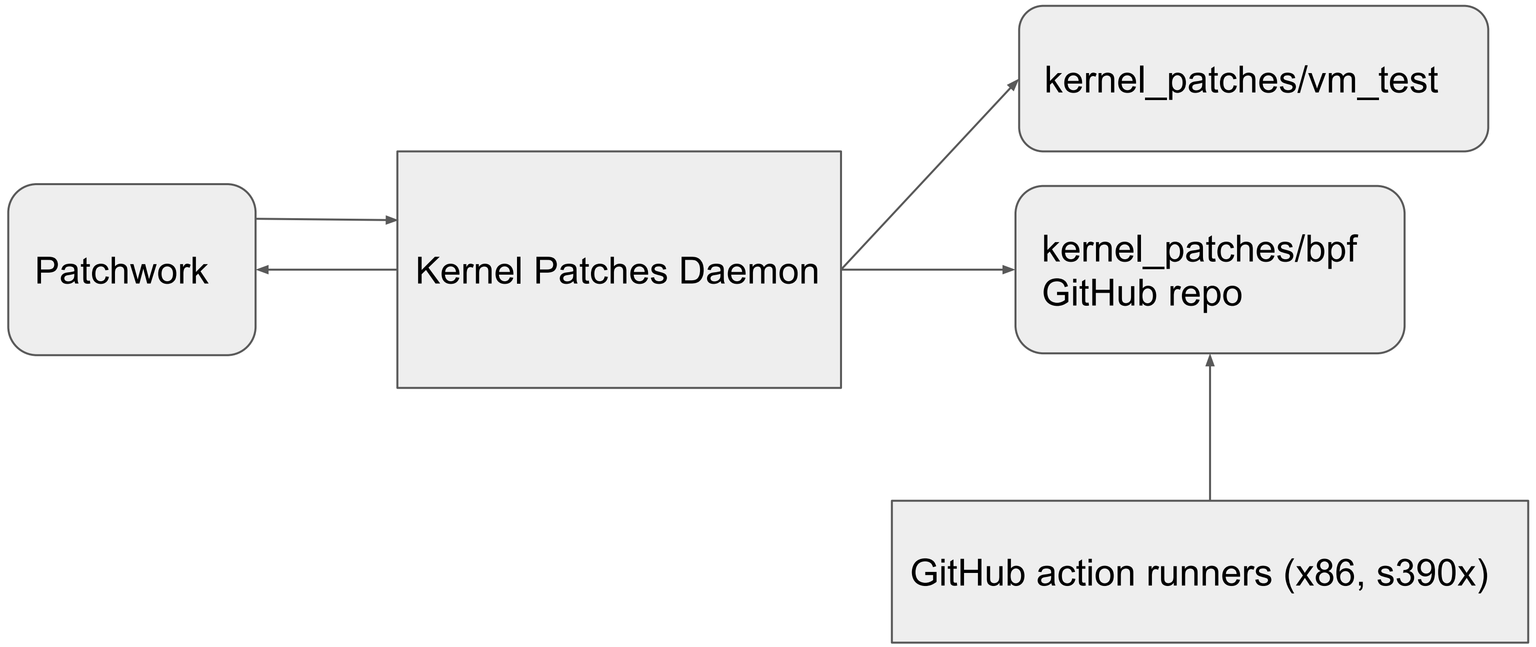


Show patches with: State = **Action Required** | Archived = **No** | 82 patches

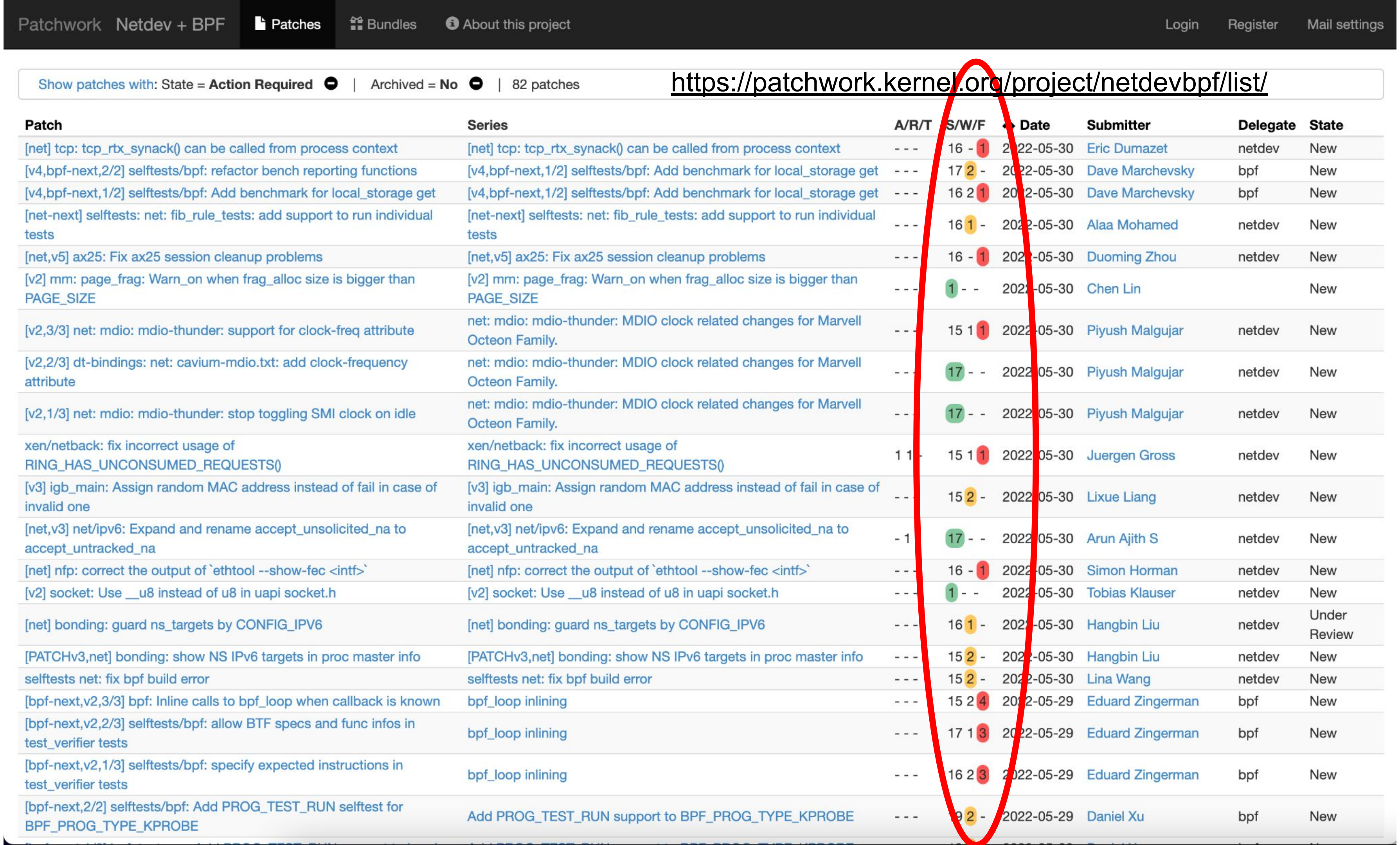
https://patchwork.kernel.org/project/netdevbpf/list/

Patch	Series	A/R/T	S/W/F	Date	Submitter	Delegate	State
[net] tcp: tcp_rtx_synack() can be called from process context	[net] tcp: tcp_rtx_synack() can be called from process context	- - -	16 - 1	2022-05-30	Eric Dumazet	netdev	New
[v4,bpf-next,2/2] selftests/bpf: refactor bench reporting functions	[v4,bpf-next,1/2] selftests/bpf: Add benchmark for local_storage get	- - -	17 2 -	2022-05-30	Dave Marchevsky	bpf	New
[v4,bpf-next,1/2] selftests/bpf: Add benchmark for local_storage get	[v4,bpf-next,1/2] selftests/bpf: Add benchmark for local_storage get	- - -	16 2 1	2022-05-30	Dave Marchevsky	bpf	New
[net-next] selftests: net: fib_rule_tests: add support to run individual tests	[net-next] selftests: net: fib_rule_tests: add support to run individual tests	- - -	16 1 -	2022-05-30	Alaa Mohamed	netdev	New
[net,v5] ax25: Fix ax25 session cleanup problems	[net,v5] ax25: Fix ax25 session cleanup problems	- - -	16 - 1	2022-05-30	Duoming Zhou	netdev	New
[v2] mm: page_frag: Warn_on when frag_alloc size is bigger than PAGE_SIZE	[v2] mm: page_frag: Warn_on when frag_alloc size is bigger than PAGE_SIZE	- - -	1 - -	2022-05-30	Chen Lin		New
[v2,3/3] net: mdio: mdio-thunder: support for clock-freq attribute	net: mdio: mdio-thunder: MDIO clock related changes for Marvell Octeon Family.	- - -	15 1 1	2022-05-30	Piyush Malgujar	netdev	New
[v2,2/3] dt-bindings: net: cavium-mdio.txt: add clock-frequency attribute	net: mdio: mdio-thunder: MDIO clock related changes for Marvell Octeon Family.	- - -	17 - -	2022-05-30	Piyush Malgujar	netdev	New
[v2,1/3] net: mdio: mdio-thunder: stop toggling SMI clock on idle	net: mdio: mdio-thunder: MDIO clock related changes for Marvell Octeon Family.	- - -	17 - -	2022-05-30	Piyush Malgujar	netdev	New
xen/netback: fix incorrect usage of RING_HAS_UNCONSUMED_REQUESTS()	xen/netback: fix incorrect usage of RING_HAS_UNCONSUMED_REQUESTS()	1 1 -	15 1 1	2022-05-30	Juergen Gross	netdev	New
[v3] igb_main: Assign random MAC address instead of fail in case of invalid one	[v3] igb_main: Assign random MAC address instead of fail in case of invalid one	- - -	15 2 -	2022-05-30	Lixue Liang	netdev	New
[net,v3] net/ipv6: Expand and rename accept_unsolicited_na to accept_untracked_na	[net,v3] net/ipv6: Expand and rename accept_unsolicited_na to accept_untracked_na	- 1 -	17 - -	2022-05-30	Arun Ajith S	netdev	New
[net] nfp: correct the output of `ethtool --show-fec <intf>`	[net] nfp: correct the output of `ethtool --show-fec <intf>`	- - -	16 - 1	2022-05-30	Simon Horman	netdev	New
[v2] socket: Use __u8 instead of u8 in uapi socket.h	[v2] socket: Use __u8 instead of u8 in uapi socket.h	- - -	1 - -	2022-05-30	Tobias Klauser	netdev	New
[net] bonding: guard ns_targets by CONFIG_IPV6	[net] bonding: guard ns_targets by CONFIG_IPV6	- - -	16 1 -	2022-05-30	Hangbin Liu	netdev	Under Review
[PATCHv3,net] bonding: show NS IPv6 targets in proc master info	[PATCHv3,net] bonding: show NS IPv6 targets in proc master info	- - -	15 2 -	2022-05-30	Hangbin Liu	netdev	New
selftests net: fix bpf build error	selftests net: fix bpf build error	- - -	15 2 -	2022-05-30	Lina Wang	netdev	New
[bpf-next,v2,3/3] bpf: Inline calls to bpf_loop when callback is known	bpf_loop inlining	- - -	15 2 4	2022-05-29	Eduard Zingerman	bpf	New
[bpf-next,v2,2/3] selftests/bpf: allow BTF specs and func infos in test_verifier tests	bpf_loop inlining	- - -	17 1 3	2022-05-29	Eduard Zingerman	bpf	New
[bpf-next,v2,1/3] selftests/bpf: specify expected instructions in test_verifier tests	bpf_loop inlining	- - -	16 2 3	2022-05-29	Eduard Zingerman	bpf	New
[bpf-next,2/2] selftests/bpf: Add PROG_TEST_RUN selftest for BPF_PROG_TYPE_KPROBE	Add PROG_TEST_RUN support to BPF_PROG_TYPE_KPROBE	- - -	19 2 -	2022-05-29	Daniel Xu	bpf	New

Components


Patchwork
Kernel Patches Daemon
kernel_patches/vm_test
kernel_patches/bpf GitHub repo
GitHub action runners (x86, s390x)

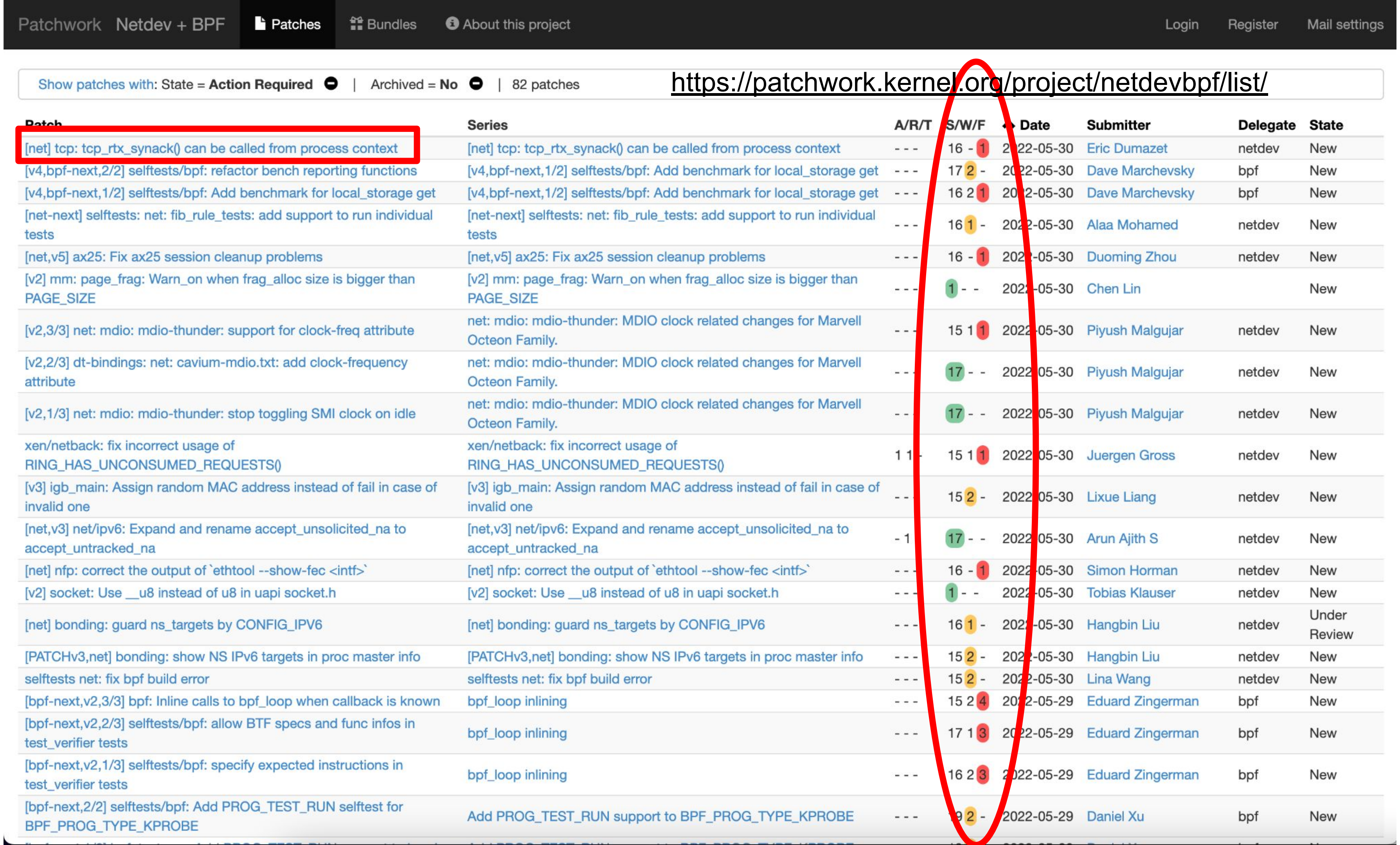
Slide copied almost verbatim from BPF CI talk by Mykola Lysenko at LSFMM 2022 (https://docs.google.com/presentation/d/1RQZjLkbXmSFOr_4Sj5BdQsXbUh_vMshXi7w09pUpWsY/edit#slide=id.g127798017a6_0_194)



Show patches with: State = **Action Required** | Archived = **No** | 82 patches

https://patchwork.kernel.org/project/netdevbpf/list/

Patch	Series	A/R/T	S/W/F	Date	Submitter	Delegate	State
[net] tcp: tcp_rtx_synack() can be called from process context	[net] tcp: tcp_rtx_synack() can be called from process context	- - -	16 - 1	2022-05-30	Eric Dumazet	netdev	New
[v4,bpf-next,2/2] selftests/bpf: refactor bench reporting functions	[v4,bpf-next,1/2] selftests/bpf: Add benchmark for local_storage get	- - -	17 2 -	2022-05-30	Dave Marchevsky	bpf	New
[v4,bpf-next,1/2] selftests/bpf: Add benchmark for local_storage get	[v4,bpf-next,1/2] selftests/bpf: Add benchmark for local_storage get	- - -	16 2 1	2022-05-30	Dave Marchevsky	bpf	New
[net-next] selftests: net: fib_rule_tests: add support to run individual tests	[net-next] selftests: net: fib_rule_tests: add support to run individual tests	- - -	16 1 -	2022-05-30	Alaa Mohamed	netdev	New
[net,v5] ax25: Fix ax25 session cleanup problems	[net,v5] ax25: Fix ax25 session cleanup problems	- - -	16 - 1	2022-05-30	Duoming Zhou	netdev	New
[v2] mm: page_frag: Warn_on when frag_alloc size is bigger than PAGE_SIZE	[v2] mm: page_frag: Warn_on when frag_alloc size is bigger than PAGE_SIZE	- - -	1 - -	2022-05-30	Chen Lin		New
[v2,3/3] net: mdio: mdio-thunder: support for clock-freq attribute	net: mdio: mdio-thunder: MDIO clock related changes for Marvell Octeon Family.	- - -	15 1 1	2022-05-30	Piyush Malgujar	netdev	New
[v2,2/3] dt-bindings: net: cavium-mdio.txt: add clock-frequency attribute	net: mdio: mdio-thunder: MDIO clock related changes for Marvell Octeon Family.	- - -	17 - -	2022-05-30	Piyush Malgujar	netdev	New
[v2,1/3] net: mdio: mdio-thunder: stop toggling SMI clock on idle	net: mdio: mdio-thunder: MDIO clock related changes for Marvell Octeon Family.	- - -	17 - -	2022-05-30	Piyush Malgujar	netdev	New
xen/netback: fix incorrect usage of RING_HAS_UNCONSUMED_REQUESTS()	xen/netback: fix incorrect usage of RING_HAS_UNCONSUMED_REQUESTS()	1 1	15 1 1	2022-05-30	Juergen Gross	netdev	New
[v3] igb_main: Assign random MAC address instead of fail in case of invalid one	[v3] igb_main: Assign random MAC address instead of fail in case of invalid one	- - -	15 2 -	2022-05-30	Lixue Liang	netdev	New
[net,v3] net/ipv6: Expand and rename accept_unsolicited_na to accept_untracked_na	[net,v3] net/ipv6: Expand and rename accept_unsolicited_na to accept_untracked_na	- 1	17 - -	2022-05-30	Arun Ajith S	netdev	New
[net] nfp: correct the output of `ethtool --show-fec <intf>`	[net] nfp: correct the output of `ethtool --show-fec <intf>`	- - -	16 - 1	2022-05-30	Simon Horman	netdev	New
[v2] socket: Use __u8 instead of u8 in uapi socket.h	[v2] socket: Use __u8 instead of u8 in uapi socket.h	- - -	1 - -	2022-05-30	Tobias Klauser	netdev	New
[net] bonding: guard ns_targets by CONFIG_IPV6	[net] bonding: guard ns_targets by CONFIG_IPV6	- - -	16 1 -	2022-05-30	Hangbin Liu	netdev	Under Review
[PATCHv3,net] bonding: show NS IPv6 targets in proc master info	[PATCHv3,net] bonding: show NS IPv6 targets in proc master info	- - -	15 2 -	2022-05-30	Hangbin Liu	netdev	New
selftests net: fix bpf build error	selftests net: fix bpf build error	- - -	15 2 -	2022-05-30	Lina Wang	netdev	New
[bpf-next,v2,3/3] bpf: Inline calls to bpf_loop when callback is known	bpf_loop inlining	- - -	15 2 4	2022-05-29	Eduard Zingerman	bpf	New
[bpf-next,v2,2/3] selftests/bpf: allow BTF specs and func infos in test_verifier tests	bpf_loop inlining	- - -	17 1 3	2022-05-29	Eduard Zingerman	bpf	New
[bpf-next,v2,1/3] selftests/bpf: specify expected instructions in test_verifier tests	bpf_loop inlining	- - -	16 2 3	2022-05-29	Eduard Zingerman	bpf	New
[bpf-next,2/2] selftests/bpf: Add PROG_TEST_RUN selftest for BPF_PROG_TYPE_KPROBE	Add PROG_TEST_RUN support to BPF_PROG_TYPE_KPROBE	- - -	19 2 -	2022-05-29	Daniel Xu	bpf	New

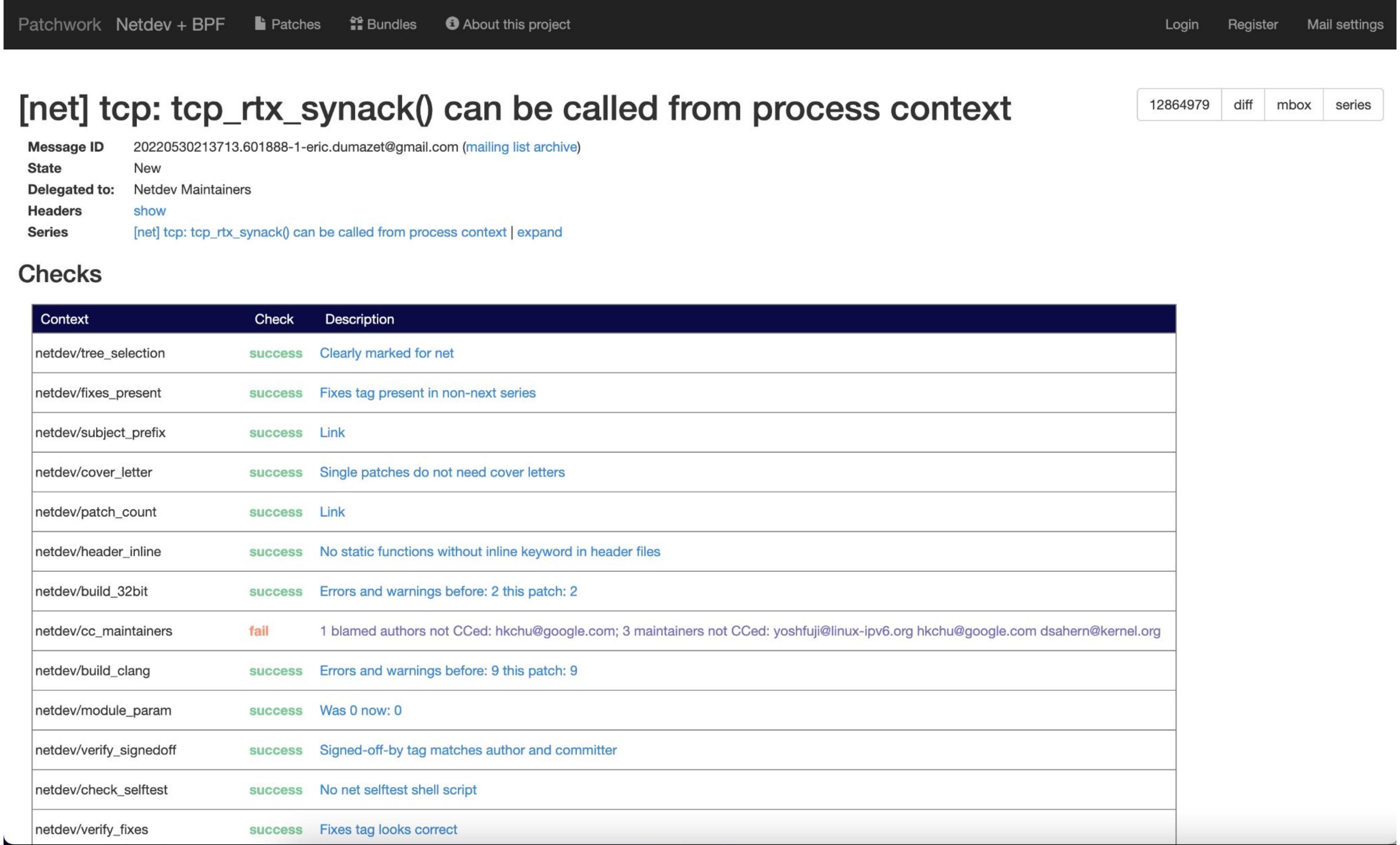


Patchwork Netdev + BPF | Patches | Bundles | About this project | Login | Register | Mail settings

Show patches with: State = **Action Required** | Archived = **No** | 82 patches

https://patchwork.kernel.org/project/netdevbpf/list/

Patch	Series	A/R/T	S/W/F	Date	Submitter	Delegate	State
[net] tcp: tcp_rtx_synack() can be called from process context	[net] tcp: tcp_rtx_synack() can be called from process context	- - -	16 - 1	2022-05-30	Eric Dumazet	netdev	New
[v4,bpf-next,2/2] selftests/bpf: refactor bench reporting functions	[v4,bpf-next,1/2] selftests/bpf: Add benchmark for local_storage get	- - -	17 2 -	2022-05-30	Dave Marchevsky	bpf	New
[v4,bpf-next,1/2] selftests/bpf: Add benchmark for local_storage get	[v4,bpf-next,1/2] selftests/bpf: Add benchmark for local_storage get	- - -	16 2 1	2022-05-30	Dave Marchevsky	bpf	New
[net-next] selftests: net: fib_rule_tests: add support to run individual tests	[net-next] selftests: net: fib_rule_tests: add support to run individual tests	- - -	16 1 -	2022-05-30	Alaa Mohamed	netdev	New
[net,v5] ax25: Fix ax25 session cleanup problems	[net,v5] ax25: Fix ax25 session cleanup problems	- - -	16 - 1	2022-05-30	Duoming Zhou	netdev	New
[v2] mm: page_frag: Warn_on when frag_alloc size is bigger than PAGE_SIZE	[v2] mm: page_frag: Warn_on when frag_alloc size is bigger than PAGE_SIZE	- - -	1 - -	2022-05-30	Chen Lin		New
[v2,3/3] net: mdio: mdio-thunder: support for clock-freq attribute	net: mdio: mdio-thunder: MDIO clock related changes for Marvell Octeon Family.	- - -	15 1 1	2022-05-30	Piyush Malgujar	netdev	New
[v2,2/3] dt-bindings: net: cavium-mdio.txt: add clock-frequency attribute	net: mdio: mdio-thunder: MDIO clock related changes for Marvell Octeon Family.	- - -	17 - -	2022-05-30	Piyush Malgujar	netdev	New
[v2,1/3] net: mdio: mdio-thunder: stop toggling SMI clock on idle	net: mdio: mdio-thunder: MDIO clock related changes for Marvell Octeon Family.	- - -	17 - -	2022-05-30	Piyush Malgujar	netdev	New
xen/netback: fix incorrect usage of RING_HAS_UNCONSUMED_REQUESTS()	xen/netback: fix incorrect usage of RING_HAS_UNCONSUMED_REQUESTS()	1 1 -	15 1 1	2022-05-30	Juergen Gross	netdev	New
[v3] igb_main: Assign random MAC address instead of fail in case of invalid one	[v3] igb_main: Assign random MAC address instead of fail in case of invalid one	- - -	15 2 -	2022-05-30	Lixue Liang	netdev	New
[net,v3] net/ipv6: Expand and rename accept_unsolicited_na to accept_untracked_na	[net,v3] net/ipv6: Expand and rename accept_unsolicited_na to accept_untracked_na	- 1 -	17 - -	2022-05-30	Arun Ajith S	netdev	New
[net] nfp: correct the output of `ethtool --show-fec <intf>`	[net] nfp: correct the output of `ethtool --show-fec <intf>`	- - -	16 - 1	2022-05-30	Simon Horman	netdev	New
[v2] socket: Use __u8 instead of u8 in uapi socket.h	[v2] socket: Use __u8 instead of u8 in uapi socket.h	- - -	1 - -	2022-05-30	Tobias Klauser	netdev	New
[net] bonding: guard ns_targets by CONFIG_IPV6	[net] bonding: guard ns_targets by CONFIG_IPV6	- - -	16 1 -	2022-05-30	Hangbin Liu	netdev	Under Review
[PATCHv3,net] bonding: show NS IPv6 targets in proc master info	[PATCHv3,net] bonding: show NS IPv6 targets in proc master info	- - -	15 2 -	2022-05-30	Hangbin Liu	netdev	New
selftests net: fix bpf build error	selftests net: fix bpf build error	- - -	15 2 -	2022-05-30	Lina Wang	netdev	New
[bpf-next,v2,3/3] bpf: Inline calls to bpf_loop when callback is known	bpf_loop inlining	- - -	15 2 4	2022-05-29	Eduard Zingerman	bpf	New
[bpf-next,v2,2/3] selftests/bpf: allow BTF specs and func infos in test_verifier tests	bpf_loop inlining	- - -	17 1 3	2022-05-29	Eduard Zingerman	bpf	New
[bpf-next,v2,1/3] selftests/bpf: specify expected instructions in test_verifier tests	bpf_loop inlining	- - -	16 2 3	2022-05-29	Eduard Zingerman	bpf	New
[bpf-next,2/2] selftests/bpf: Add PROG_TEST_RUN selftest for BPF_PROG_TYPE_KPROBE	Add PROG_TEST_RUN support to BPF_PROG_TYPE_KPROBE	- - -	19 2 -	2022-05-29	Daniel Xu	bpf	New



Patchwork Netdev + BPF | Patches | Bundles | About this project | Login | Register | Mail settings

[net] tcp: tcp_rtx_synack() can be called from process context

12864979 | diff | mbox | series

Message ID	20220530213713.601888-1-eric.dumazet@gmail.com (mailing list archive)
State	New
Delegated to:	Netdev Maintainers
Headers	show
Series	[net] tcp: tcp_rtx_synack() can be called from process context | expand

Checks

Context	Check	Description
netdev/tree_selection	success	Clearly marked for net
netdev/fixes_present	success	Fixes tag present in non-next series
netdev/subject_prefix	success	Link
netdev/cover_letter	success	Single patches do not need cover letters
netdev/patch_count	success	Link
netdev/header_inline	success	No static functions without inline keyword in header files
netdev/build_32bit	success	Errors and warnings before: 2 this patch: 2
netdev/cc_maintainers	fail	1 blamed authors not CCed: hkchu@google.com; 3 maintainers not CCed: yoshfuji@linux-ipv6.org hkchu@google.com dsahern@kernel.org
netdev/build_clang	success	Errors and warnings before: 9 this patch: 9
netdev/module_param	success	Was 0 now: 0
netdev/verify_signedoff	success	Signed-off-by tag matches author and committer
netdev/check_selftest	success	No net selftest shell script
netdev/verify_fixes	success	Fixes tag looks correct

Index of /static/nipa/646089/12864979/cc_maintainers/

```
../
desc                                               30-May-2022 21:45                 129
retcode                                            30-May-2022 21:45                   1
summary                                            30-May-2022 21:45                  36
```

```
==========
cc_maintainers - FAILED
```

Patchwork

Pros

- Patchwork is used by maintainers (one stop shops can be nice)
- Runs on every patch sent to BPF lists
- Runs on at least 2 architectures, could theoretically add more
- BPF tests in general are easy to run locally – can use script to run in VM
- New BPF tests automatically run

Cons

- Other patchwork suites need their own daemon, etc infra to run CI
- Doesn’t send messages to BPF lists for job failures
- Uses Meta / private infrastructure for Kernel Patches daemon
- Doesn’t run tests on SoCs or directly on various non-x86 hardware (uses QEMU for s390x)

Meta

Bonus: Other CI options

Linux* Kernel Performance

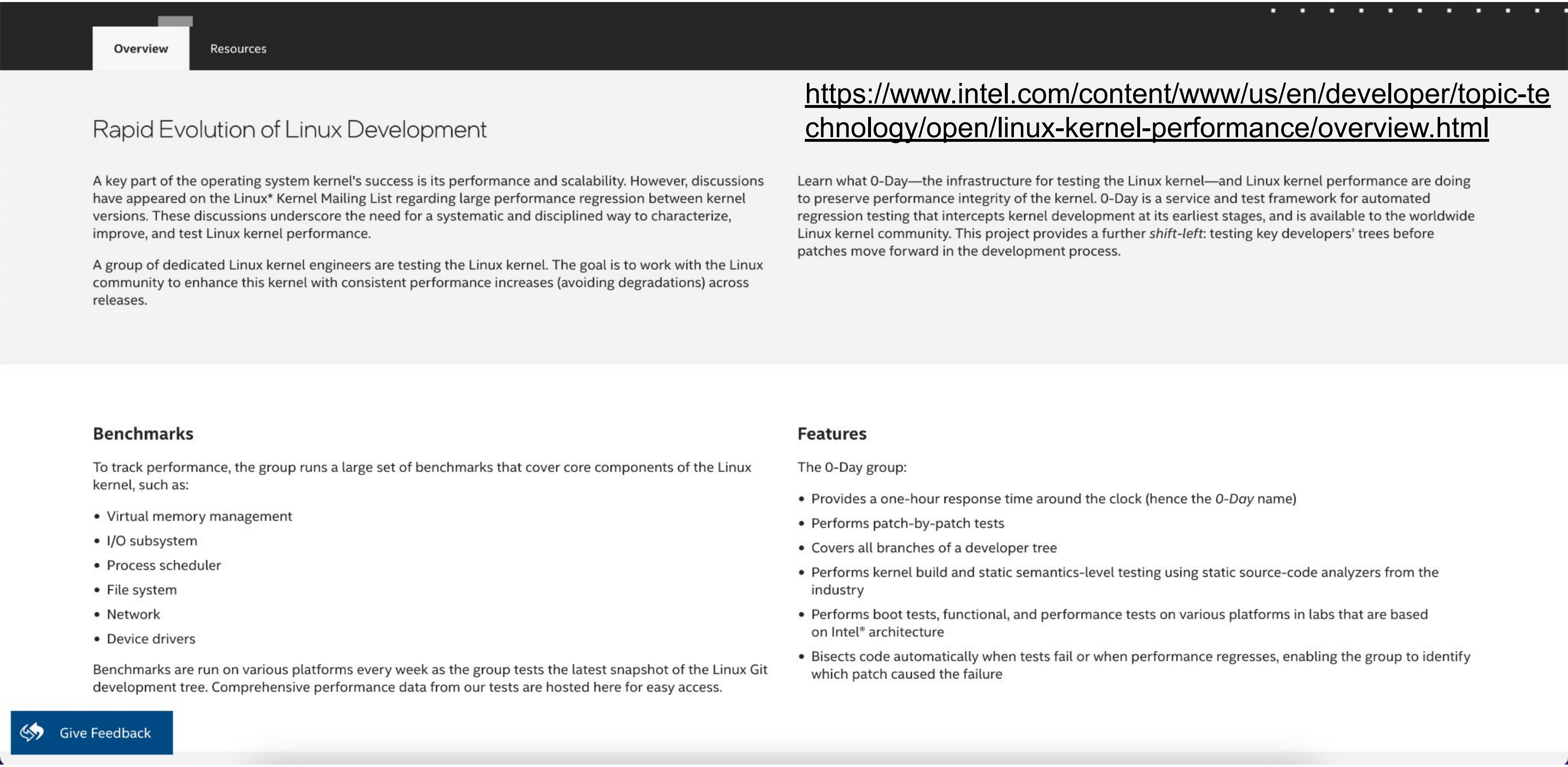
LKP – Linux Kernel Performance / 0 day

Run by the 0-day team at Intel

Builds and runs kernels across a variety of trees, branches, toolchains, and configs, including unmerged patches

Runs build tests, benchmarks, and logical tests (defined out of tree in separate github repo)

Only builds and tests on and for x86 (though apparently they also build for other architectures on private jobs / branches?)



https://www.intel.com/content/www/us/en/developer/topic-technology/open/linux-kernel-performance/overview.html

Rapid Evolution of Linux Development

A key part of the operating system kernel's success is its performance and scalability. However, discussions have appeared on the Linux* Kernel Mailing List regarding large performance regression between kernel versions. These discussions underscore the need for a systematic and disciplined way to characterize, improve, and test Linux kernel performance.

A group of dedicated Linux kernel engineers are testing the Linux kernel. The goal is to work with the Linux community to enhance this kernel with consistent performance increases (avoiding degradations) across releases.

Learn what 0-Day—the infrastructure for testing the Linux kernel—and Linux kernel performance are doing to preserve performance integrity of the kernel. 0-Day is a service and test framework for automated regression testing that intercepts kernel development at its earliest stages, and is available to the worldwide Linux kernel community. This project provides a further *shift-left*: testing key developers' trees before patches move forward in the development process.

Benchmarks

To track performance, the group runs a large set of benchmarks that cover core components of the Linux kernel, such as:

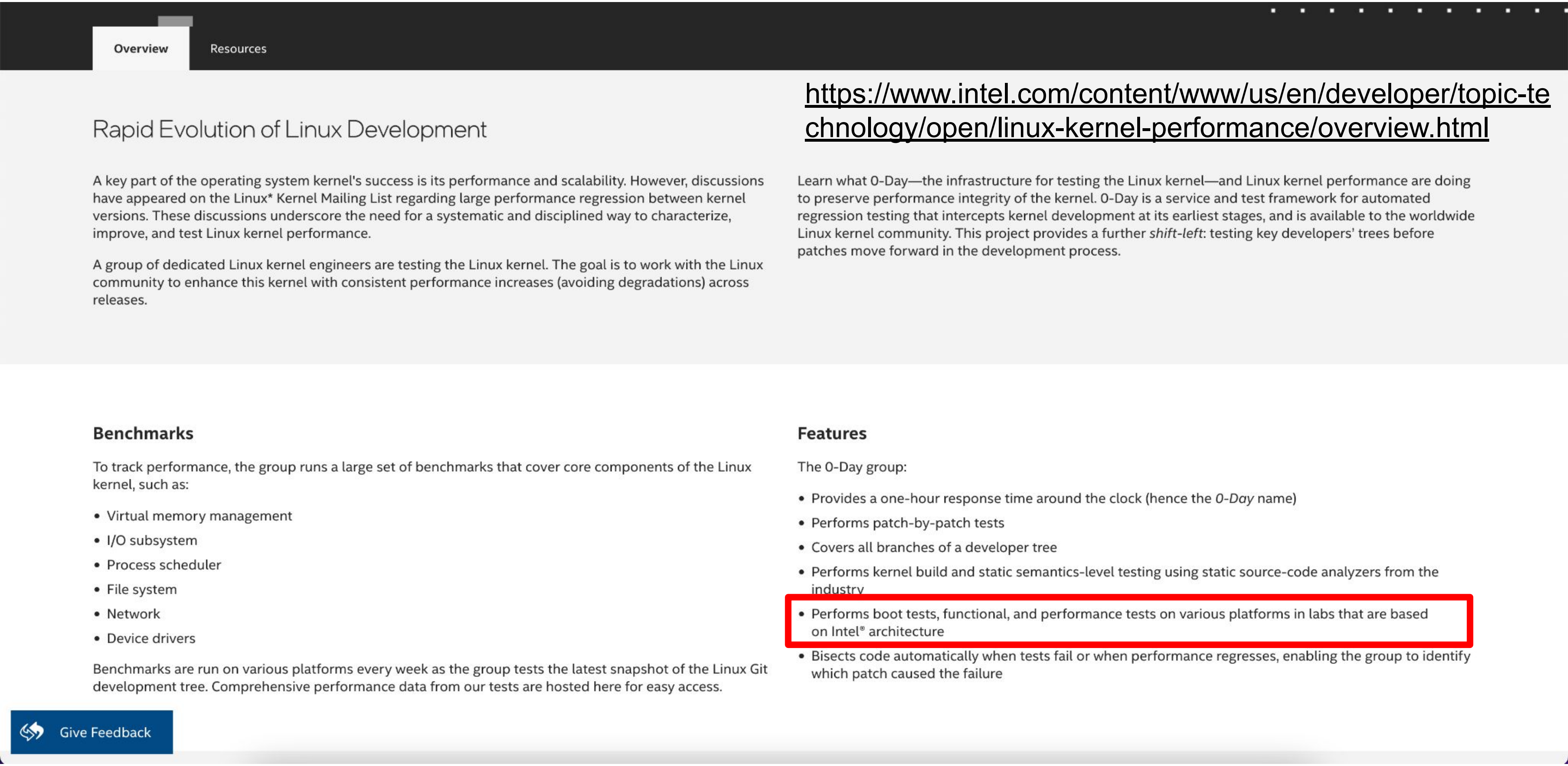
- Virtual memory management
- I/O subsystem
- Process scheduler
- File system
- Network
- Device drivers

Benchmarks are run on various platforms every week as the group tests the latest snapshot of the Linux Git development tree. Comprehensive performance data from our tests are hosted here for easy access.

Features

The 0-Day group:

- Provides a one-hour response time around the clock (hence the *0-Day* name)
- Performs patch-by-patch tests
- Covers all branches of a developer tree
- Performs kernel build and static semantics-level testing using static source-code analyzers from the industry
- Performs boot tests, functional, and performance tests on various platforms in labs that are based on Intel® architecture
- Bisects code automatically when tests fail or when performance regresses, enabling the group to identify which patch caused the failure



Overview Resources

https://www.intel.com/content/www/us/en/developer/topic-technology/open/linux-kernel-performance/overview.html

Rapid Evolution of Linux Development

A key part of the operating system kernel's success is its performance and scalability. However, discussions have appeared on the Linux* Kernel Mailing List regarding large performance regression between kernel versions. These discussions underscore the need for a systematic and disciplined way to characterize, improve, and test Linux kernel performance.

A group of dedicated Linux kernel engineers are testing the Linux kernel. The goal is to work with the Linux community to enhance this kernel with consistent performance increases (avoiding degradations) across releases.

Learn what 0-Day—the infrastructure for testing the Linux kernel—and Linux kernel performance are doing to preserve performance integrity of the kernel. 0-Day is a service and test framework for automated regression testing that intercepts kernel development at its earliest stages, and is available to the worldwide Linux kernel community. This project provides a further *shift-left*: testing key developers' trees before patches move forward in the development process.

Benchmarks

To track performance, the group runs a large set of benchmarks that cover core components of the Linux kernel, such as:

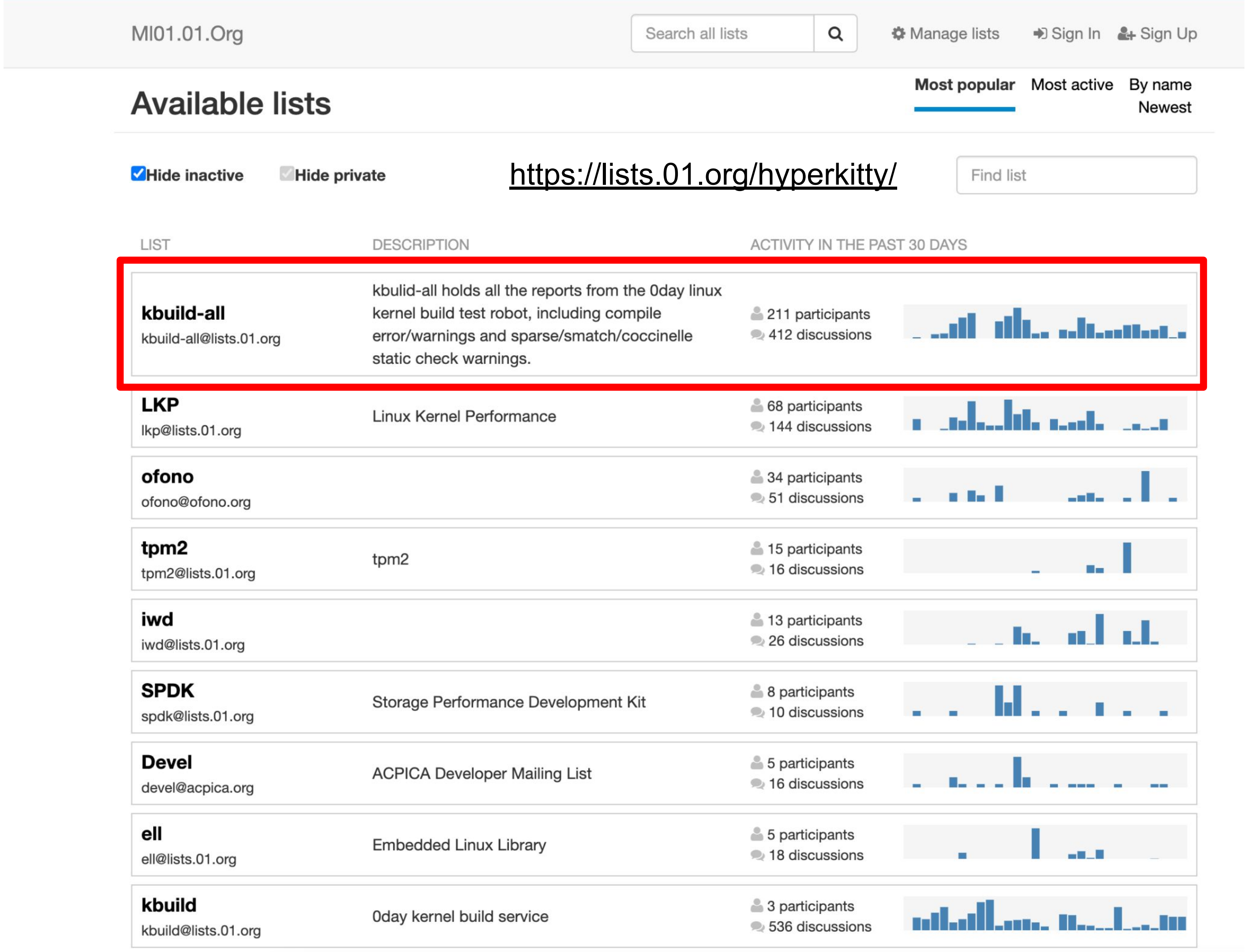
- Virtual memory management
- I/O subsystem
- Process scheduler
- File system
- Network
- Device drivers

Benchmarks are run on various platforms every week as the group tests the latest snapshot of the Linux Git development tree. Comprehensive performance data from our tests are hosted here for easy access.

Features

The 0-Day group:

- Provides a one-hour response time around the clock (hence the *0-Day* name)
- Performs patch-by-patch tests
- Covers all branches of a developer tree
- Performs kernel build and static semantics-level testing using static source-code analyzers from the industry
- Performs boot tests, functional, and performance tests on various platforms in labs that are based on Intel® architecture
- Bisects code automatically when tests fail or when performance regresses, enabling the group to identify which patch caused the failure



MI01.01.Org

Search all lists

Manage lists · Sign In · Sign Up

Available lists

Most popular · Most active · By name · Newest

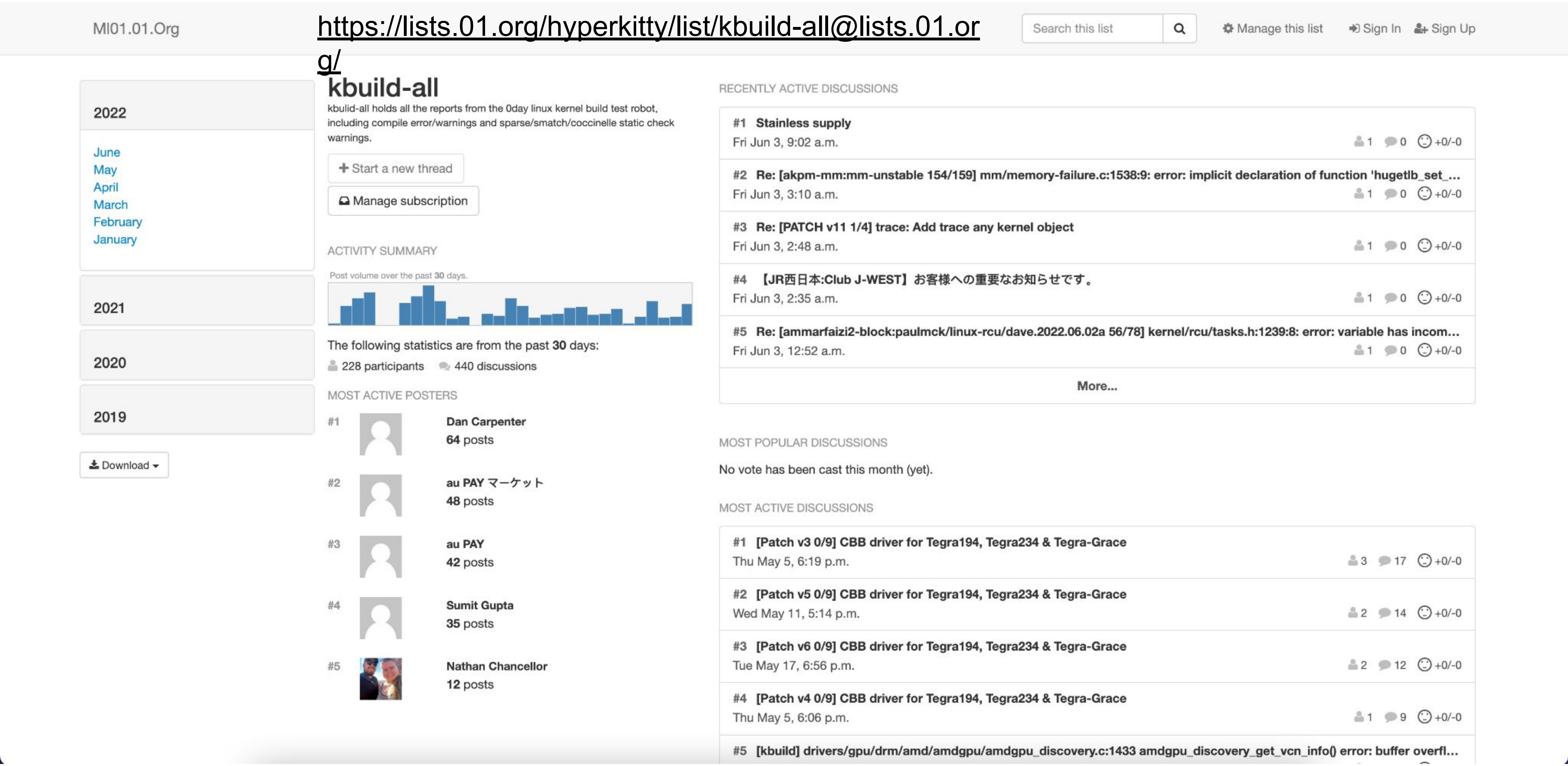
☑ Hide inactive ☑ Hide private

https://lists.01.org/hyperkitty/

Find list

LIST	DESCRIPTION	ACTIVITY IN THE PAST 30 DAYS
kbuild-all kbuild-all@lists.01.org	kbulid-all holds all the reports from the 0day linux kernel build test robot, including compile error/warnings and sparse/smatch/coccinelle static check warnings.	211 participants 412 discussions
LKP lkp@lists.01.org	Linux Kernel Performance	68 participants 144 discussions
ofono ofono@ofono.org		34 participants 51 discussions
tpm2 tpm2@lists.01.org	tpm2	15 participants 16 discussions
iwd iwd@lists.01.org		13 participants 26 discussions
SPDK spdk@lists.01.org	Storage Performance Development Kit	8 participants 10 discussions
Devel devel@acpica.org	ACPICA Developer Mailing List	5 participants 16 discussions
ell ell@lists.01.org	Embedded Linux Library	5 participants 18 discussions
kbuild kbuild@lists.01.org	0day kernel build service	3 participants 536 discussions

https://lists.01.org/hyperkitty/list/kbuild-all@lists.01.org/



Ml01.01.Org

Search this list | Manage this list | Sign In | Sign Up

2022

June
May
April
March
February
January

2021

2020

2019

Download

kbuild-all

kbulid-all holds all the reports from the 0day linux kernel build test robot, including compile error/warnings and sparse/smatch/coccinelle static check warnings.

Start a new thread

Manage subscription

ACTIVITY SUMMARY

Post volume over the past **30** days.

The following statistics are from the past **30** days:

228 participants 440 discussions

MOST ACTIVE POSTERS

#1	Dan Carpenter	**64** posts
#2	au PAY マーケット	**48** posts
#3	au PAY	**42** posts
#4	Sumit Gupta	**35** posts
#5	Nathan Chancellor	**12** posts

RECENTLY ACTIVE DISCUSSIONS

#1	**Stainless supply** Fri Jun 3, 9:02 a.m.	1	0	+0/-0
#2	**Re: [akpm-mm:mm-unstable 154/159] mm/memory-failure.c:1538:9: error: implicit declaration of function 'hugetlb_set_...** Fri Jun 3, 3:10 a.m.	1	0	+0/-0
#3	**Re: [PATCH v11 1/4] trace: Add trace any kernel object** Fri Jun 3, 2:48 a.m.	1	0	+0/-0
#4	**【JR西日本:Club J-WEST】お客様への重要なお知らせです。** Fri Jun 3, 2:35 a.m.	1	0	+0/-0
#5	**Re: [ammarfaizi2-block:paulmck/linux-rcu/dave.2022.06.02a 56/78] kernel/rcu/tasks.h:1239:8: error: variable has incom...** Fri Jun 3, 12:52 a.m.	1	0	+0/-0

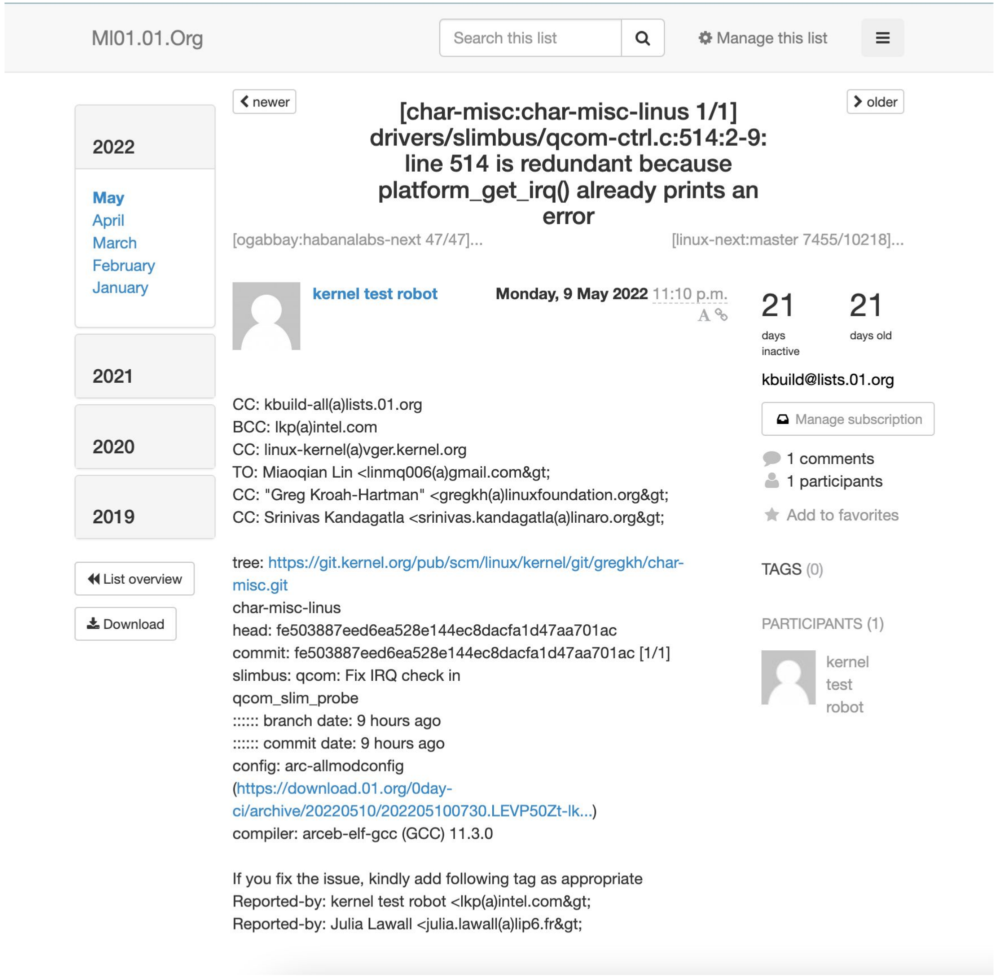
More...

MOST POPULAR DISCUSSIONS

No vote has been cast this month (yet).

MOST ACTIVE DISCUSSIONS

#1	**[Patch v3 0/9] CBB driver for Tegra194, Tegra234 & Tegra-Grace** Thu May 5, 6:19 p.m.	3	17	+0/-0
#2	**[Patch v5 0/9] CBB driver for Tegra194, Tegra234 & Tegra-Grace** Wed May 11, 5:14 p.m.	2	14	+0/-0
#3	**[Patch v6 0/9] CBB driver for Tegra194, Tegra234 & Tegra-Grace** Tue May 17, 6:56 p.m.	2	12	+0/-0
#4	**[Patch v4 0/9] CBB driver for Tegra194, Tegra234 & Tegra-Grace** Thu May 5, 6:06 p.m.	1	9	+0/-0
#5	**[kbuild] drivers/gpu/drm/amd/amdgpu/amdgpu_discovery.c:1433 amdgpu_discovery_get_vcn_info() error: buffer overfl...**			



MI01.01.Org

Search this list | Manage this list

< newer | > older

[char-misc:char-misc-linus 1/1] drivers/slimbus/qcom-ctrl.c:514:2-9: line 514 is redundant because platform_get_irq() already prints an error

[ogabbay:habanalabs-next 47/47]... [linux-next:master 7455/10218]...

2022: May, April, March, February, January
2021
2020
2019

List overview
Download

kernel test robot — Monday, 9 May 2022 11:10 p.m.

21 days inactive — 21 days old

kbuild@lists.01.org

Manage subscription
1 comments
1 participants
Add to favorites

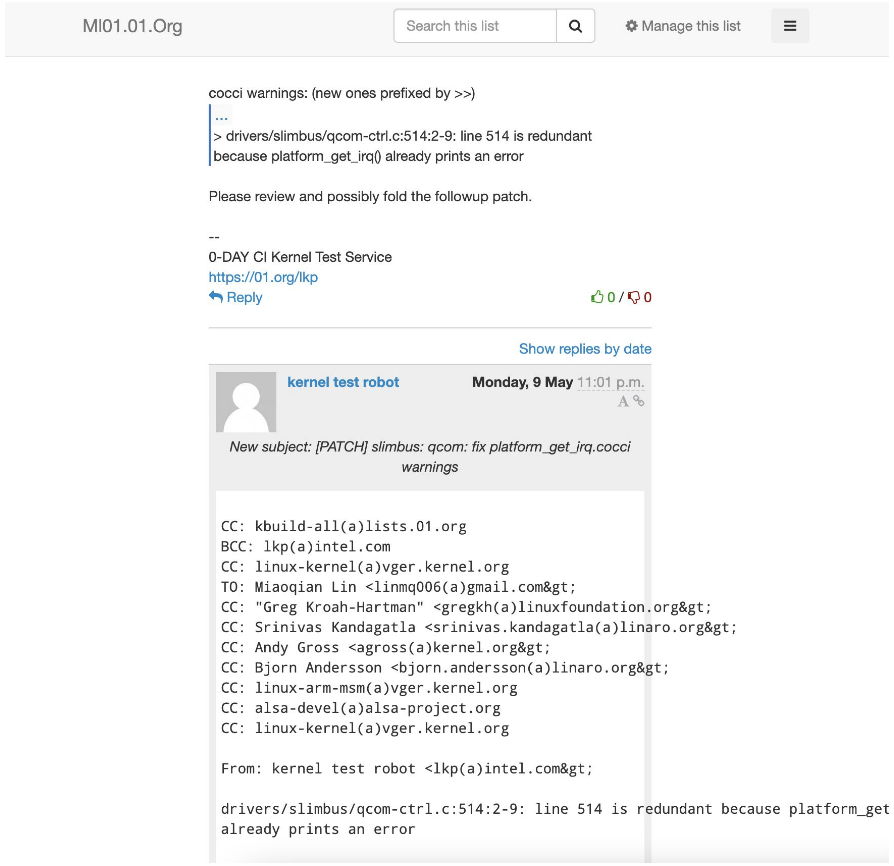
TAGS (0)

PARTICIPANTS (1)
kernel test robot

CC: kbuild-all(a)lists.01.org
BCC: lkp(a)intel.com
CC: linux-kernel(a)vger.kernel.org
TO: Miaoqian Lin <linmq006(a)gmail.com&gt;
CC: "Greg Kroah-Hartman" <gregkh(a)linuxfoundation.org&gt;
CC: Srinivas Kandagatla <srinivas.kandagatla(a)linaro.org&gt;

tree: https://git.kernel.org/pub/scm/linux/kernel/git/gregkh/char-misc.git
char-misc-linus
head: fe503887eed6ea528e144ec8dacfa1d47aa701ac
commit: fe503887eed6ea528e144ec8dacfa1d47aa701ac [1/1]
slimbus: qcom: Fix IRQ check in
qcom_slim_probe
:::::: branch date: 9 hours ago
:::::: commit date: 9 hours ago
config: arc-allmodconfig
(https://download.01.org/0day-ci/archive/20220510/202205100730.LEVP50Zt-lk...)
compiler: arceb-elf-gcc (GCC) 11.3.0

If you fix the issue, kindly add following tag as appropriate
Reported-by: kernel test robot <lkp(a)intel.com&gt;
Reported-by: Julia Lawall <julia.lawall(a)lip6.fr&gt;



MI01.01.Org

Search this list | Manage this list

cocci warnings: (new ones prefixed by >>)
...
> drivers/slimbus/qcom-ctrl.c:514:2-9: line 514 is redundant because platform_get_irq() already prints an error

Please review and possibly fold the followup patch.

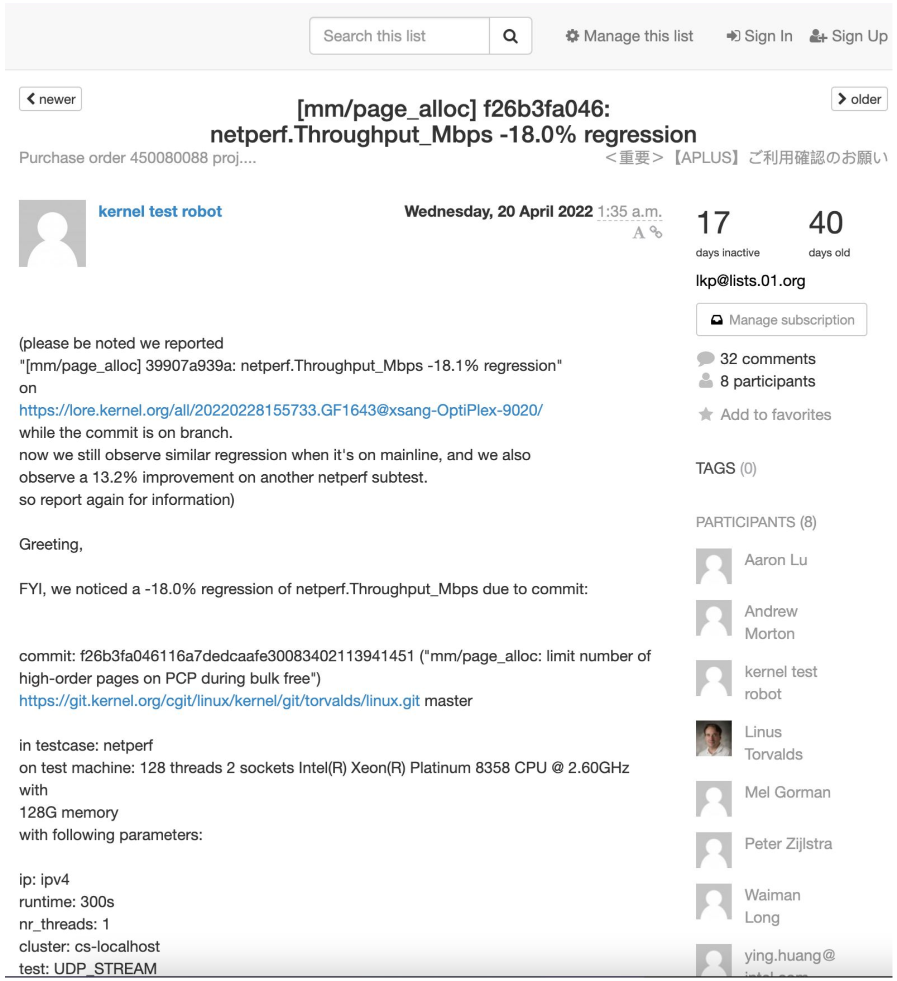
--
0-DAY CI Kernel Test Service
https://01.org/lkp
Reply 0 / 0

Show replies by date

kernel test robot — Monday, 9 May 11:01 p.m.

New subject: [PATCH] slimbus: qcom: fix platform_get_irq.cocci warnings

```
CC: kbuild-all(a)lists.01.org
BCC: lkp(a)intel.com
CC: linux-kernel(a)vger.kernel.org
TO: Miaoqian Lin <linmq006(a)gmail.com&gt;
CC: "Greg Kroah-Hartman" <gregkh(a)linuxfoundation.org&gt;
CC: Srinivas Kandagatla <srinivas.kandagatla(a)linaro.org&gt;
CC: Andy Gross <agross(a)kernel.org&gt;
CC: Bjorn Andersson <bjorn.andersson(a)linaro.org&gt;
CC: linux-arm-msm(a)vger.kernel.org
CC: alsa-devel(a)alsa-project.org
CC: linux-kernel(a)vger.kernel.org

From: kernel test robot <lkp(a)intel.com&gt;

drivers/slimbus/qcom-ctrl.c:514:2-9: line 514 is redundant because platform_get
already prints an error
```



[mm/page_alloc] f26b3fa046: netperf.Throughput_Mbps -18.0% regression

Purchase order 450080088 proj.... ＜重要＞【APLUS】ご利用確認のお願い

kernel test robot **Wednesday, 20 April 2022** 1:35 a.m.

17 days inactive 40 days old

lkp@lists.01.org

32 comments
8 participants
Add to favorites

TAGS (0)

PARTICIPANTS (8)

Aaron Lu
Andrew Morton
kernel test robot
Linus Torvalds
Mel Gorman
Peter Zijlstra
Waiman Long
ying.huang@

(please be noted we reported
"[mm/page_alloc] 39907a939a: netperf.Throughput_Mbps -18.1% regression"
on
https://lore.kernel.org/all/20220228155733.GF1643@xsang-OptiPlex-9020/
while the commit is on branch.
now we still observe similar regression when it's on mainline, and we also
observe a 13.2% improvement on another netperf subtest.
so report again for information)

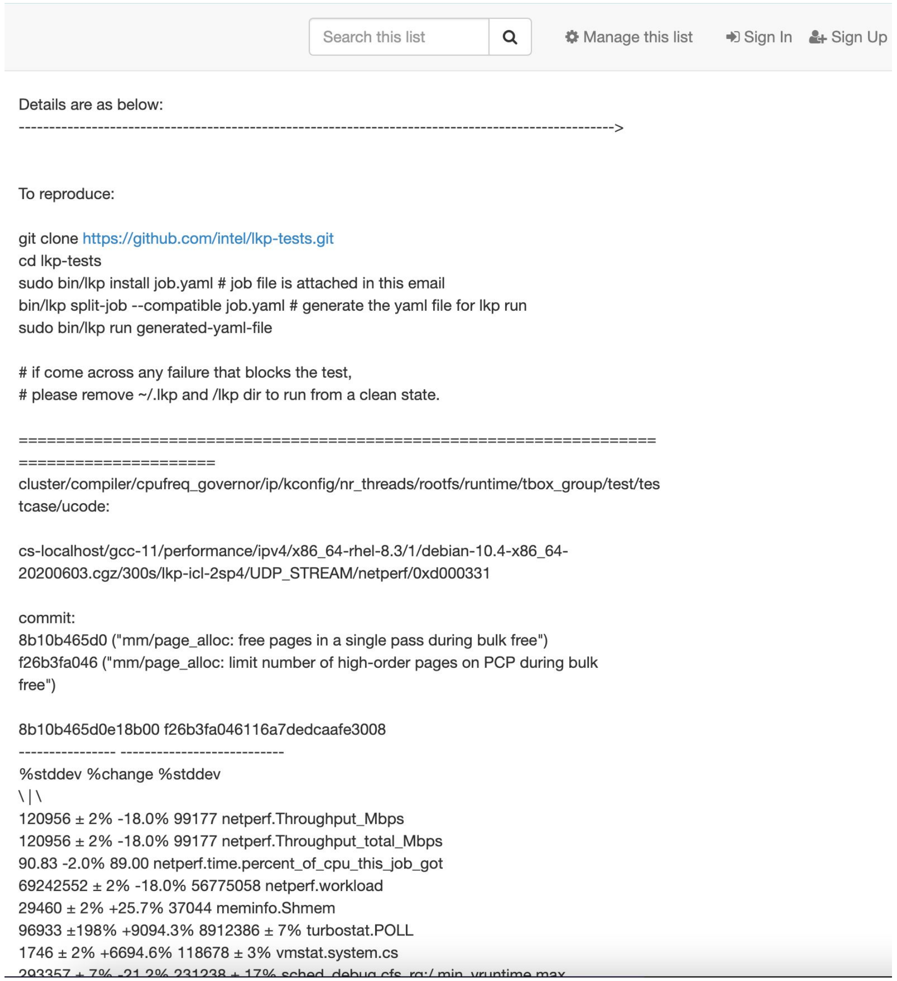
Greeting,

FYI, we noticed a -18.0% regression of netperf.Throughput_Mbps due to commit:

commit: f26b3fa046116a7dedcaafe30083402113941451 ("mm/page_alloc: limit number of high-order pages on PCP during bulk free")
https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git master

in testcase: netperf
on test machine: 128 threads 2 sockets Intel(R) Xeon(R) Platinum 8358 CPU @ 2.60GHz with
128G memory
with following parameters:

ip: ipv4
runtime: 300s
nr_threads: 1
cluster: cs-localhost
test: UDP_STREAM



Details are as below:
-->

To reproduce:

```
git clone https://github.com/intel/lkp-tests.git
cd lkp-tests
sudo bin/lkp install job.yaml # job file is attached in this email
bin/lkp split-job --compatible job.yaml # generate the yaml file for lkp run
sudo bin/lkp run generated-yaml-file

# if come across any failure that blocks the test,
# please remove ~/.lkp and /lkp dir to run from a clean state.
```

```
=========================================================================================
cluster/compiler/cpufreq_governor/ip/kconfig/nr_threads/rootfs/runtime/tbox_group/test/tes
tcase/ucode:

cs-localhost/gcc-11/performance/ipv4/x86_64-rhel-8.3/1/debian-10.4-x86_64-
20200603.cgz/300s/lkp-icl-2sp4/UDP_STREAM/netperf/0xd000331

commit:
8b10b465d0 ("mm/page_alloc: free pages in a single pass during bulk free")
f26b3fa046 ("mm/page_alloc: limit number of high-order pages on PCP during bulk
free")

8b10b465d0e18b00 f26b3fa046116a7dedcaafe3008
---------------- ---------------------------
%stddev %change %stddev
\ | \
120956 ± 2% -18.0% 99177 netperf.Throughput_Mbps
120956 ± 2% -18.0% 99177 netperf.Throughput_total_Mbps
90.83 -2.0% 89.00 netperf.time.percent_of_cpu_this_job_got
69242552 ± 2% -18.0% 56775058 netperf.workload
29460 ± 2% +25.7% 37044 meminfo.Shmem
96933 ±198% +9094.3% 8912386 ± 7% turbostat.POLL
1746 ± 2% +6694.6% 118678 ± 3% vmstat.system.cs
293357 ± 7% -21.2% 231238 ± 17% sched_debug.cfs_rq:/.min_vruntime.max
```

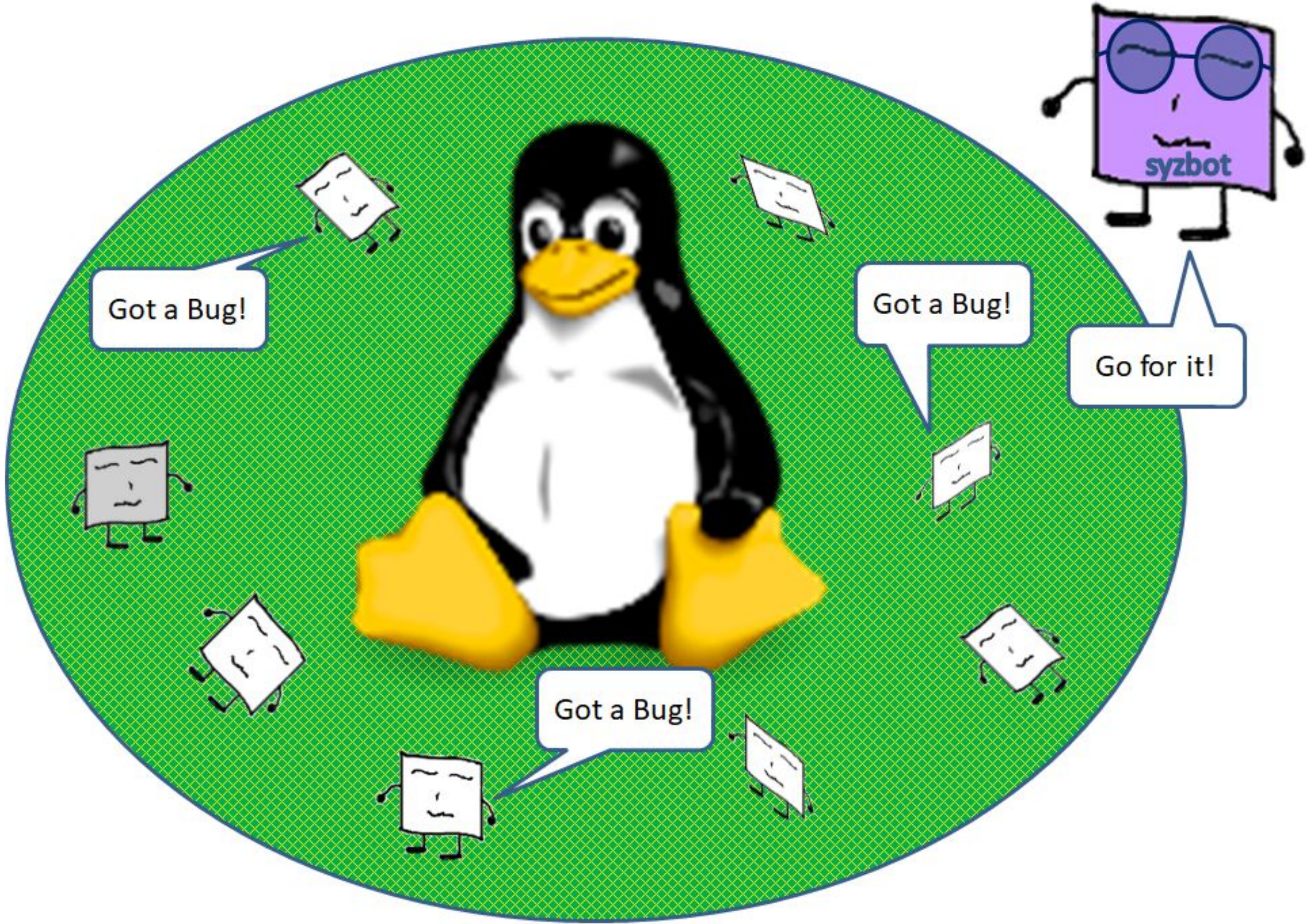
LKP / 0 Day – Pros and Cons

Pros

- Builds on patches that have not yet been merged
- Provides strong signal by sending messages to upstream lists
- Runs benchmarks
- Does bisection to find initial broken commit

Cons

- Only runs builds and tests for x86 (or not?)
- Does not build with multiple toolchains
- Error information helpful, but less comprehensive than KernelCI
- Uses Intel / private infrastructure (and source?)


syzbot
Got a Bug!
Got a Bug!
Go for it!
Got a Bug!

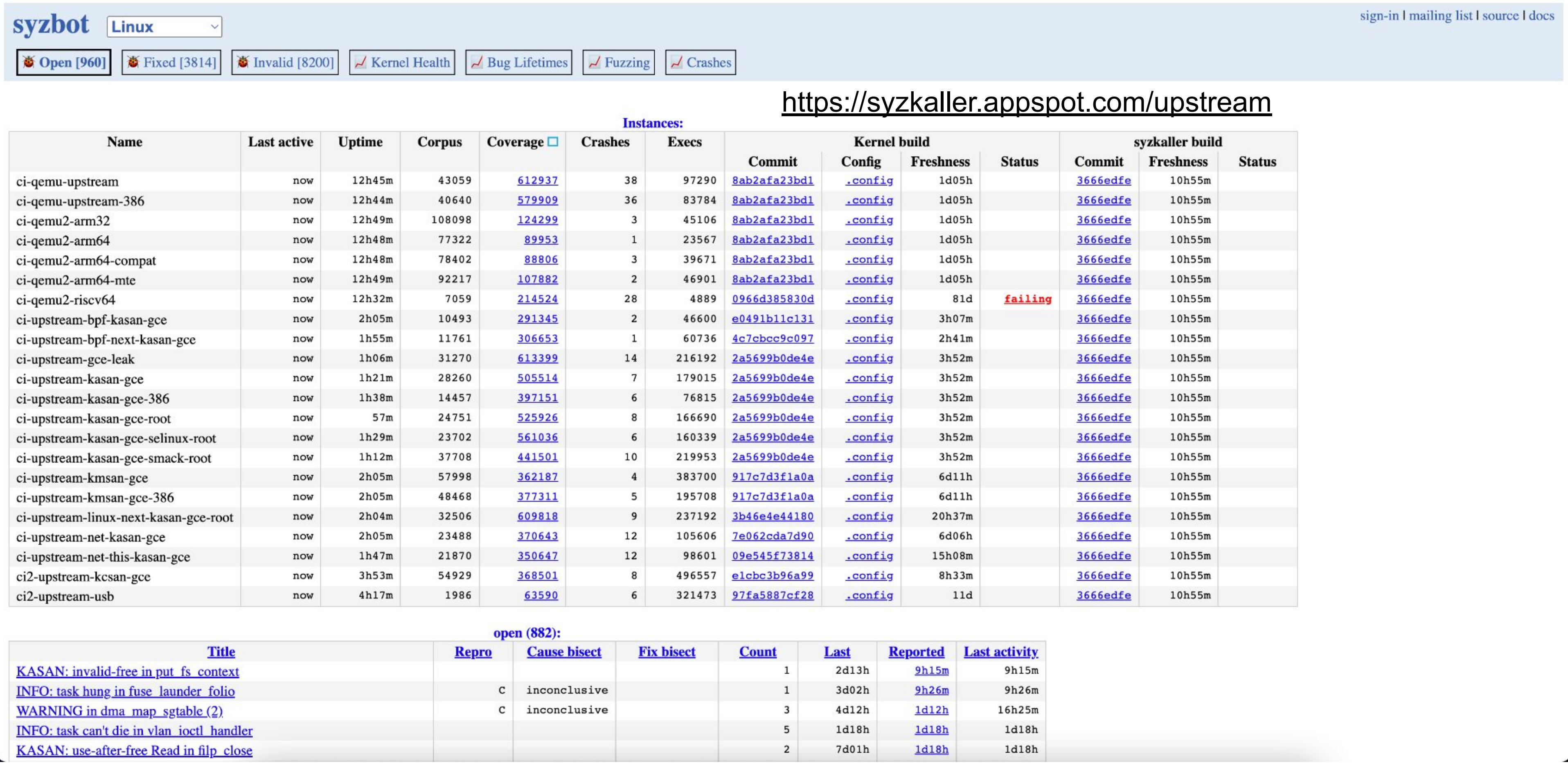
syzkaller + syzbot – Fuzzing the kernel

Continuously fuzzes main Linux kernel branches

Reports found bugs to upstream lists

Bisects to find bugs (and fixes) on specific patches

Runs on multiple architectures



syzbot Linux

sign-in | mailing list | source | docs

Open [960] | Fixed [3814] | Invalid [8200] | Kernel Health | Bug Lifetimes | Fuzzing | Crashes

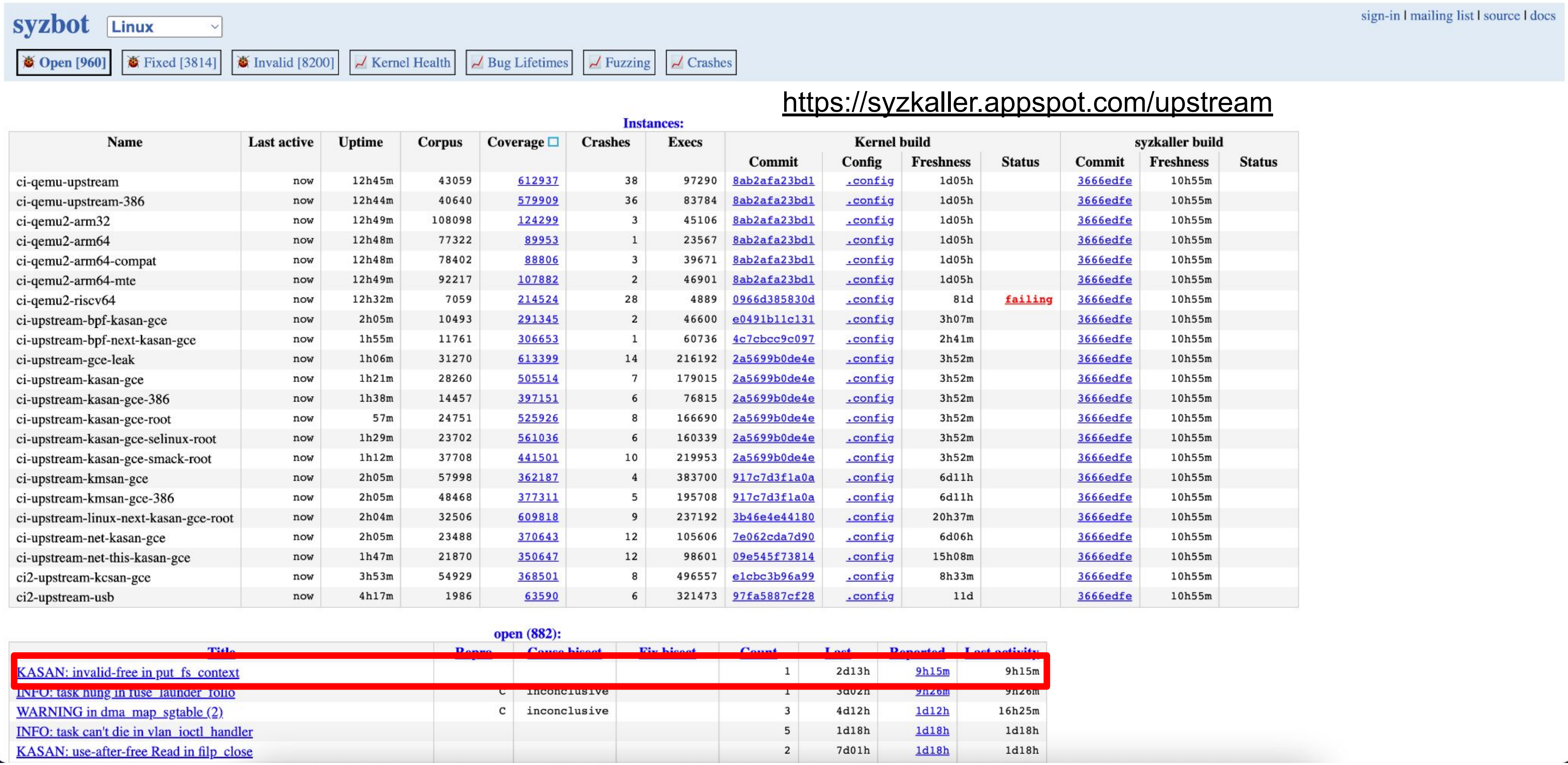
https://syzkaller.appspot.com/upstream

Instances:

Name	Last active	Uptime	Corpus	Coverage	Crashes	Execs	Kernel build				syzkaller build		
							Commit	Config	Freshness	Status	Commit	Freshness	Status
ci-qemu-upstream	now	12h45m	43059	612937	38	97290	8ab2afa23bd1	.config	1d05h		3666edfe	10h55m	
ci-qemu-upstream-386	now	12h44m	40640	579909	36	83784	8ab2afa23bd1	.config	1d05h		3666edfe	10h55m	
ci-qemu2-arm32	now	12h49m	108098	124299	3	45106	8ab2afa23bd1	.config	1d05h		3666edfe	10h55m	
ci-qemu2-arm64	now	12h48m	77322	89953	1	23567	8ab2afa23bd1	.config	1d05h		3666edfe	10h55m	
ci-qemu2-arm64-compat	now	12h48m	78402	88806	3	39671	8ab2afa23bd1	.config	1d05h		3666edfe	10h55m	
ci-qemu2-arm64-mte	now	12h49m	92217	107882	2	46901	8ab2afa23bd1	.config	1d05h		3666edfe	10h55m	
ci-qemu2-riscv64	now	12h32m	7059	214524	28	4889	0966d385830d	.config	81d	failing	3666edfe	10h55m	
ci-upstream-bpf-kasan-gce	now	2h05m	10493	291345	2	46600	e0491b11c131	.config	3h07m		3666edfe	10h55m	
ci-upstream-bpf-next-kasan-gce	now	1h55m	11761	306653	1	60736	4c7cbcc9c097	.config	2h41m		3666edfe	10h55m	
ci-upstream-gce-leak	now	1h06m	31270	613399	14	216192	2a5699b0de4e	.config	3h52m		3666edfe	10h55m	
ci-upstream-kasan-gce	now	1h21m	28260	505514	7	179015	2a5699b0de4e	.config	3h52m		3666edfe	10h55m	
ci-upstream-kasan-gce-386	now	1h38m	14457	397151	6	76815	2a5699b0de4e	.config	3h52m		3666edfe	10h55m	
ci-upstream-kasan-gce-root	now	57m	24751	525926	8	166690	2a5699b0de4e	.config	3h52m		3666edfe	10h55m	
ci-upstream-kasan-gce-selinux-root	now	1h29m	23702	561036	6	160339	2a5699b0de4e	.config	3h52m		3666edfe	10h55m	
ci-upstream-kasan-gce-smack-root	now	1h12m	37708	441501	10	219953	2a5699b0de4e	.config	3h52m		3666edfe	10h55m	
ci-upstream-kmsan-gce	now	2h05m	57998	362187	4	383700	917c7d3f1a0a	.config	6d11h		3666edfe	10h55m	
ci-upstream-kmsan-gce-386	now	2h05m	48468	377311	5	195708	917c7d3f1a0a	.config	6d11h		3666edfe	10h55m	
ci-upstream-linux-next-kasan-gce-root	now	2h04m	32506	609818	9	237192	3b46e4e44180	.config	20h37m		3666edfe	10h55m	
ci-upstream-net-kasan-gce	now	2h05m	23488	370643	12	105606	7e062cda7d90	.config	6d06h		3666edfe	10h55m	
ci-upstream-net-this-kasan-gce	now	1h47m	21870	350647	12	98601	09e545f73814	.config	15h08m		3666edfe	10h55m	
ci2-upstream-kcsan-gce	now	3h53m	54929	368501	8	496557	e1cbc3b96a99	.config	8h33m		3666edfe	10h55m	
ci2-upstream-usb	now	4h17m	1986	63590	6	321473	97fa5887cf28	.config	11d		3666edfe	10h55m	

open (882):

Title	Repro	Cause bisect	Fix bisect	Count	Last	Reported	Last activity
KASAN: invalid-free in put_fs_context				1	2d13h	9h15m	9h15m
INFO: task hung in fuse_launder_folio	C	inconclusive		1	3d02h	9h26m	9h26m
WARNING in dma_map_sgtable (2)	C	inconclusive		3	4d12h	1d12h	16h25m
INFO: task can't die in vlan_ioctl_handler				5	1d18h	1d18h	1d18h
KASAN: use-after-free Read in filp_close				2	7d01h	1d18h	1d18h



syzbot Linux

sign-in | mailing list | source | docs

Open [960] | Fixed [3814] | Invalid [8200] | Kernel Health | Bug Lifetimes | Fuzzing | Crashes

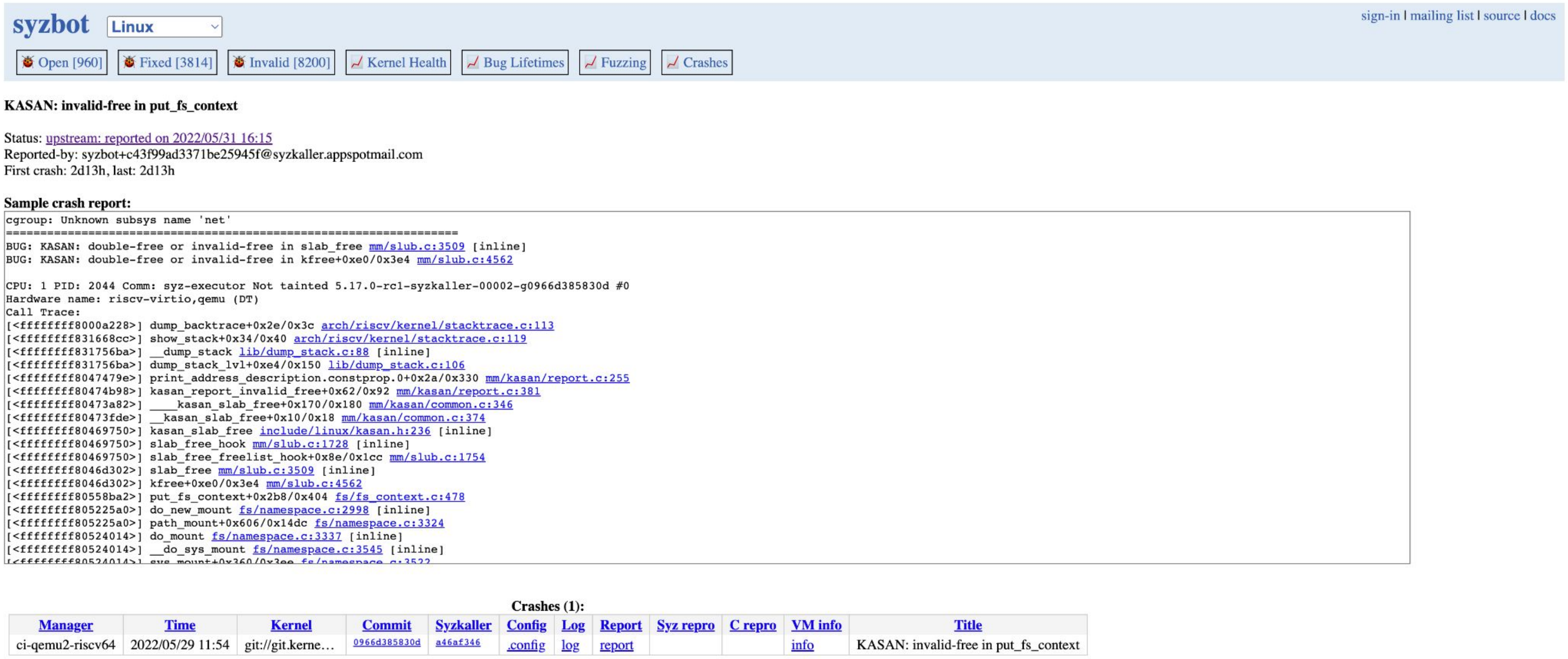
https://syzkaller.appspot.com/upstream

Instances:

Name	Last active	Uptime	Corpus	Coverage	Crashes	Execs	Kernel build				syzkaller build		
							Commit	Config	Freshness	Status	Commit	Freshness	Status
ci-qemu-upstream	now	12h45m	43059	612937	38	97290	8ab2afa23bd1	.config	1d05h		3666edfe	10h55m	
ci-qemu-upstream-386	now	12h44m	40640	579909	36	83784	8ab2afa23bd1	.config	1d05h		3666edfe	10h55m	
ci-qemu2-arm32	now	12h49m	108098	124299	3	45106	8ab2afa23bd1	.config	1d05h		3666edfe	10h55m	
ci-qemu2-arm64	now	12h48m	77322	89953	1	23567	8ab2afa23bd1	.config	1d05h		3666edfe	10h55m	
ci-qemu2-arm64-compat	now	12h48m	78402	88806	3	39671	8ab2afa23bd1	.config	1d05h		3666edfe	10h55m	
ci-qemu2-arm64-mte	now	12h49m	92217	107882	2	46901	8ab2afa23bd1	.config	1d05h		3666edfe	10h55m	
ci-qemu2-riscv64	now	12h32m	7059	214524	28	4889	0966d385830d	.config	81d	failing	3666edfe	10h55m	
ci-upstream-bpf-kasan-gce	now	2h05m	10493	291345	2	46600	e0491b11c131	.config	3h07m		3666edfe	10h55m	
ci-upstream-bpf-next-kasan-gce	now	1h55m	11761	306653	1	60736	4c7cbcc9c097	.config	2h41m		3666edfe	10h55m	
ci-upstream-gce-leak	now	1h06m	31270	613399	14	216192	2a5699b0de4e	.config	3h52m		3666edfe	10h55m	
ci-upstream-kasan-gce	now	1h21m	28260	505514	7	179015	2a5699b0de4e	.config	3h52m		3666edfe	10h55m	
ci-upstream-kasan-gce-386	now	1h38m	14457	397151	6	76815	2a5699b0de4e	.config	3h52m		3666edfe	10h55m	
ci-upstream-kasan-gce-root	now	57m	24751	525926	8	166690	2a5699b0de4e	.config	3h52m		3666edfe	10h55m	
ci-upstream-kasan-gce-selinux-root	now	1h29m	23702	561036	6	160339	2a5699b0de4e	.config	3h52m		3666edfe	10h55m	
ci-upstream-kasan-gce-smack-root	now	1h12m	37708	441501	10	219953	2a5699b0de4e	.config	3h52m		3666edfe	10h55m	
ci-upstream-kmsan-gce	now	2h05m	57998	362187	4	383700	917c7d3f1a0a	.config	6d11h		3666edfe	10h55m	
ci-upstream-kmsan-gce-386	now	2h05m	48468	377311	5	195708	917c7d3f1a0a	.config	6d11h		3666edfe	10h55m	
ci-upstream-linux-next-kasan-gce-root	now	2h04m	32506	609818	9	237192	3b46e4e44180	.config	20h37m		3666edfe	10h55m	
ci-upstream-net-kasan-gce	now	2h05m	23488	370643	12	105606	7e062cda7d90	.config	6d06h		3666edfe	10h55m	
ci-upstream-net-this-kasan-gce	now	1h47m	21870	350647	12	98601	09e545f73814	.config	15h08m		3666edfe	10h55m	
ci2-upstream-kcsan-gce	now	3h53m	54929	368501	8	496557	e1cbc3b96a99	.config	8h33m		3666edfe	10h55m	
ci2-upstream-usb	now	4h17m	1986	63590	6	321473	97fa5887cf28	.config	11d		3666edfe	10h55m	

open (882):

Title	Repro	Cause bisect	Fix bisect	Count	Last	Reported	Last activity
KASAN: invalid-free in put_fs_context				1	2d13h	9h15m	9h15m
INFO: task hung in fuse_launder_folio	C	inconclusive		1	3d02h	9h26m	9h26m
WARNING in dma_map_sgtable (2)	C	inconclusive		3	4d12h	1d12h	16h25m
INFO: task can't die in vlan_ioctl_handler				5	1d18h	1d18h	1d18h
KASAN: use-after-free Read in filp_close				2	7d01h	1d18h	1d18h



KASAN: invalid-free in put_fs_context

Status: upstream: reported on 2022/05/31 16:15
Reported-by: syzbot+c43f99ad3371be25945f@syzkaller.appspotmail.com
First crash: 2d13h, last: 2d13h

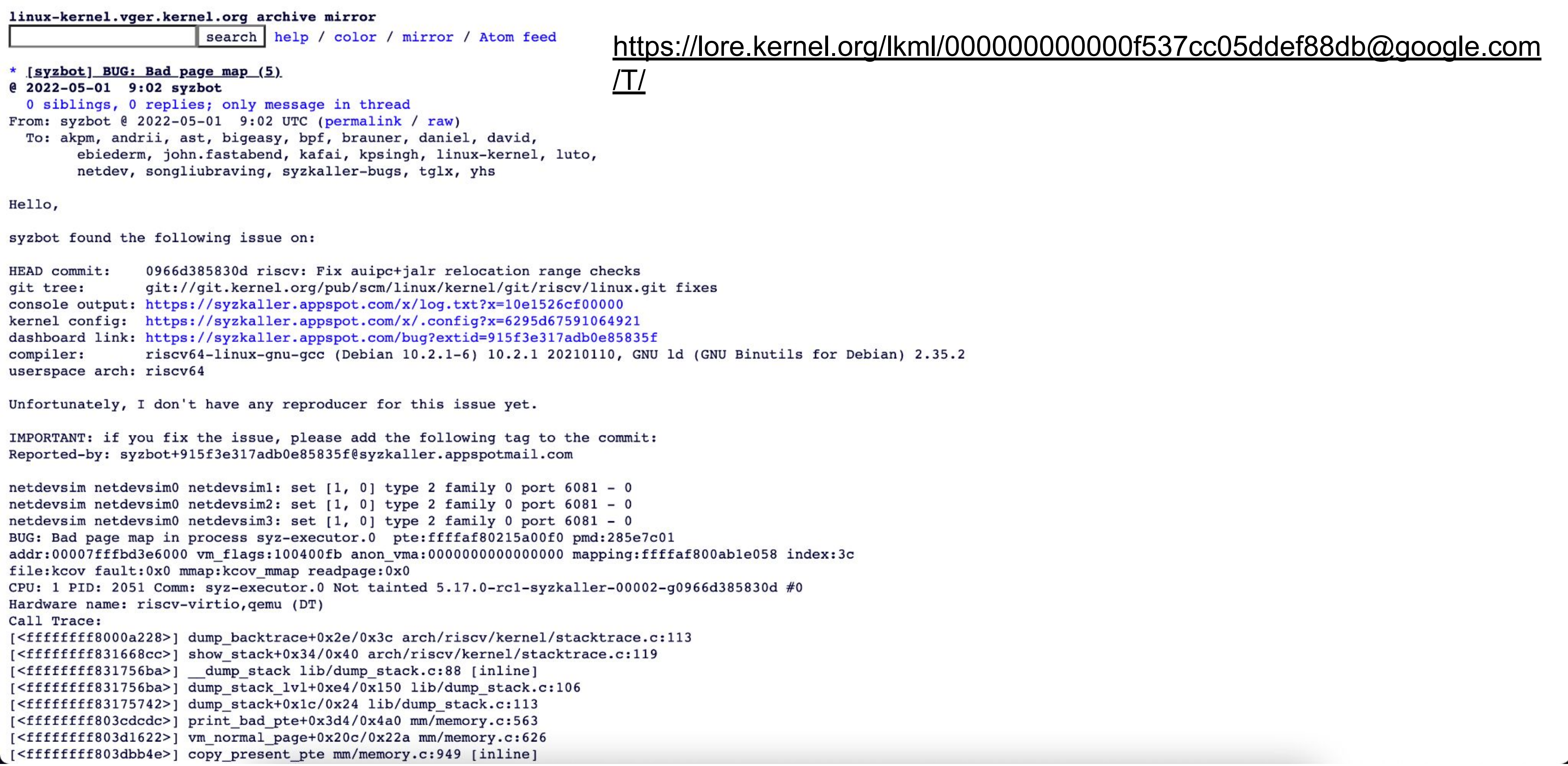
Sample crash report:

```
cgroup: Unknown subsys name 'net'
==================================================================
BUG: KASAN: double-free or invalid-free in slab_free mm/slub.c:3509 [inline]
BUG: KASAN: double-free or invalid-free in kfree+0xe0/0x3e4 mm/slub.c:4562

CPU: 1 PID: 2044 Comm: syz-executor Not tainted 5.17.0-rc1-syzkaller-00002-g0966d385830d #0
Hardware name: riscv-virtio,qemu (DT)
Call Trace:
[<ffffffff8000a228>] dump_backtrace+0x2e/0x3c arch/riscv/kernel/stacktrace.c:113
[<ffffffff831668cc>] show_stack+0x34/0x40 arch/riscv/kernel/stacktrace.c:119
[<ffffffff831756ba>] __dump_stack lib/dump_stack.c:88 [inline]
[<ffffffff831756ba>] dump_stack_lvl+0xe4/0x150 lib/dump_stack.c:106
[<ffffffff8047479e>] print_address_description.constprop.0+0x2a/0x330 mm/kasan/report.c:255
[<ffffffff80474b98>] kasan_report_invalid_free+0x62/0x92 mm/kasan/report.c:381
[<ffffffff80473a82>] ____kasan_slab_free+0x170/0x180 mm/kasan/common.c:346
[<ffffffff80473fde>] __kasan_slab_free+0x10/0x18 mm/kasan/common.c:374
[<ffffffff80469750>] kasan_slab_free include/linux/kasan.h:236 [inline]
[<ffffffff80469750>] slab_free_hook mm/slub.c:1728 [inline]
[<ffffffff80469750>] slab_free_freelist_hook+0x8e/0x1cc mm/slub.c:1754
[<ffffffff8046d302>] slab_free mm/slub.c:3509 [inline]
[<ffffffff8046d302>] kfree+0xe0/0x3e4 mm/slub.c:4562
[<ffffffff80558ba2>] put_fs_context+0x2b8/0x404 fs/fs_context.c:478
[<ffffffff805225a0>] do_new_mount fs/namespace.c:2998 [inline]
[<ffffffff805225a0>] path_mount+0x606/0x14dc fs/namespace.c:3324
[<ffffffff80524014>] do_mount fs/namespace.c:3337 [inline]
[<ffffffff80524014>] __do_sys_mount fs/namespace.c:3545 [inline]
[<ffffffff80524014>] sys_mount+0x360/0x3ee fs/namespace.c:3522
```

Crashes (1):

Manager	Time	Kernel	Commit	Syzkaller	Config	Log	Report	Syz repro	C repro	VM info	Title
ci-qemu2-riscv64	2022/05/29 11:54	git://git.kerne…	0966d385830d	a46af346	.config	log	report			info	KASAN: invalid-free in put_fs_context

https://lore.kernel.org/lkml/000000000000f537cc05ddef88db@google.com/T/



```
linux-kernel.vger.kernel.org archive mirror
[                    ] search  help / color / mirror / Atom feed

* [syzbot] BUG: Bad page map (5)
@ 2022-05-01  9:02 syzbot
  0 siblings, 0 replies; only message in thread
From: syzbot @ 2022-05-01  9:02 UTC (permalink / raw)
  To: akpm, andrii, ast, bigeasy, bpf, brauner, daniel, david,
        ebiederm, john.fastabend, kafai, kpsingh, linux-kernel, luto,
        netdev, songliubraving, syzkaller-bugs, tglx, yhs

Hello,

syzbot found the following issue on:

HEAD commit:    0966d385830d riscv: Fix auipc+jalr relocation range checks
git tree:       git://git.kernel.org/pub/scm/linux/kernel/git/riscv/linux.git fixes
console output: https://syzkaller.appspot.com/x/log.txt?x=10e1526cf00000
kernel config:  https://syzkaller.appspot.com/x/.config?x=6295d67591064921
dashboard link: https://syzkaller.appspot.com/bug?extid=915f3e317adb0e85835f
compiler:       riscv64-linux-gnu-gcc (Debian 10.2.1-6) 10.2.1 20210110, GNU ld (GNU Binutils for Debian) 2.35.2
userspace arch: riscv64

Unfortunately, I don't have any reproducer for this issue yet.

IMPORTANT: if you fix the issue, please add the following tag to the commit:
Reported-by: syzbot+915f3e317adb0e85835f@syzkaller.appspotmail.com

netdevsim netdevsim0 netdevsim1: set [1, 0] type 2 family 0 port 6081 - 0
netdevsim netdevsim0 netdevsim2: set [1, 0] type 2 family 0 port 6081 - 0
netdevsim netdevsim0 netdevsim3: set [1, 0] type 2 family 0 port 6081 - 0
BUG: Bad page map in process syz-executor.0  pte:ffffaf80215a00f0 pmd:285e7c01
addr:00007fffbd3e6000 vm_flags:100400fb anon_vma:0000000000000000 mapping:ffffaf800ab1e058 index:3c
file:kcov fault:0x0 mmap:kcov_mmap readpage:0x0
CPU: 1 PID: 2051 Comm: syz-executor.0 Not tainted 5.17.0-rc1-syzkaller-00002-g0966d385830d #0
Hardware name: riscv-virtio,qemu (DT)
Call Trace:
[<ffffffff8000a228>] dump_backtrace+0x2e/0x3c arch/riscv/kernel/stacktrace.c:113
[<ffffffff831668cc>] show_stack+0x34/0x40 arch/riscv/kernel/stacktrace.c:119
[<ffffffff831756ba>] __dump_stack lib/dump_stack.c:88 [inline]
[<ffffffff831756ba>] dump_stack_lvl+0xe4/0x150 lib/dump_stack.c:106
[<ffffffff83175742>] dump_stack+0x1c/0x24 lib/dump_stack.c:113
[<ffffffff803cdcdc>] print_bad_pte+0x3d4/0x4a0 mm/memory.c:563
[<ffffffff803d1622>] vm_normal_page+0x20c/0x22a mm/memory.c:626
[<ffffffff803dbb4e>] copy_present_pte mm/memory.c:949 [inline]
```

syzbot

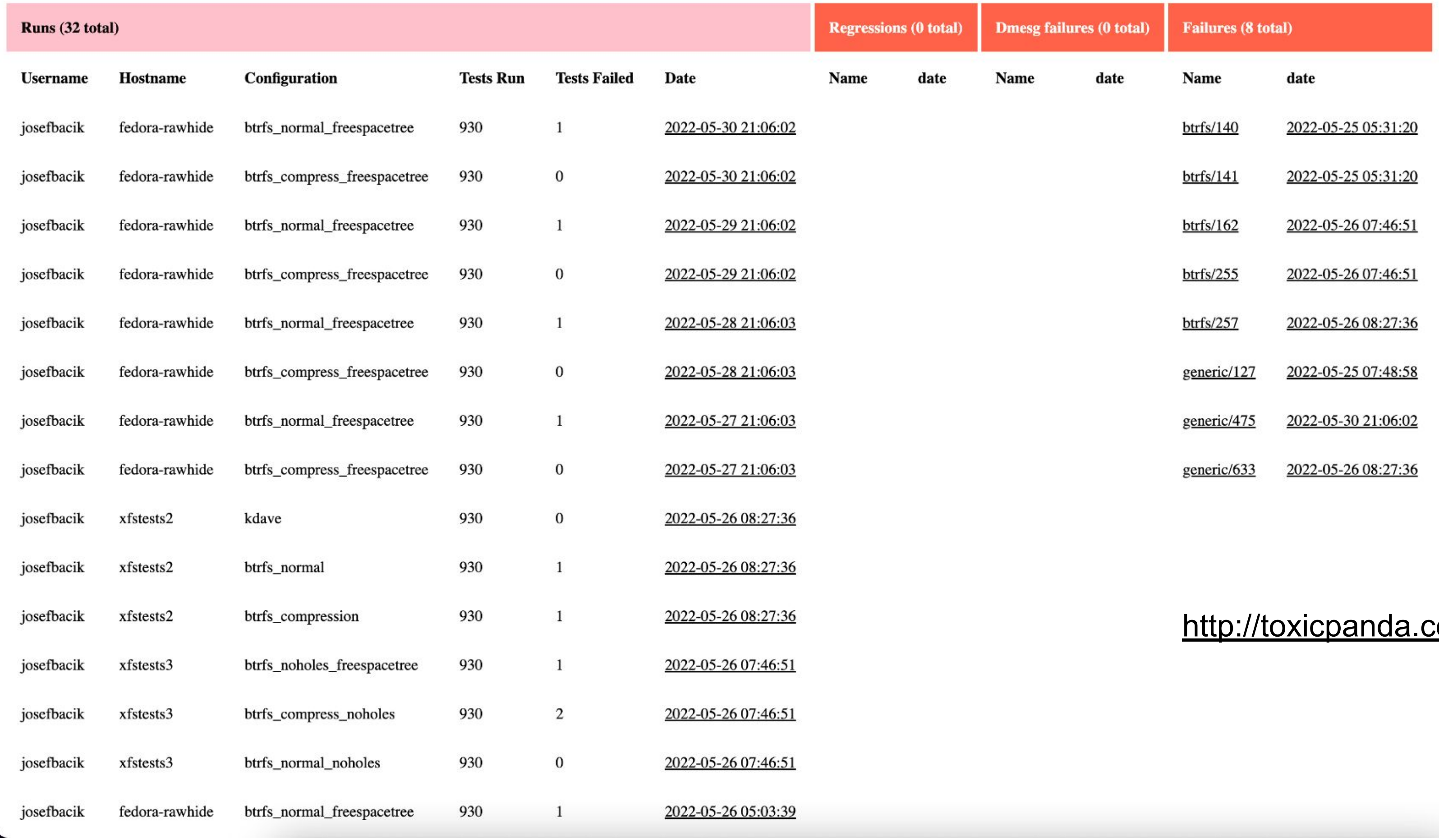
Pros

- Great coverage thanks to the nature of fuzzing + sanitizers
- Bisects to find culprit patch, and the patch that fixes an issue
- Runs on multiple architectures (in VMs)
- Sends messages to upstream on failures

Cons

- Doesn’t run on unmerged patches
- Doesn’t run selftests / kunit tests
- Runs on proprietary Google infra
- Configurations are hard-coded per platform in the syzbot repo

Independently managed solutions (e.g. for btrfs)



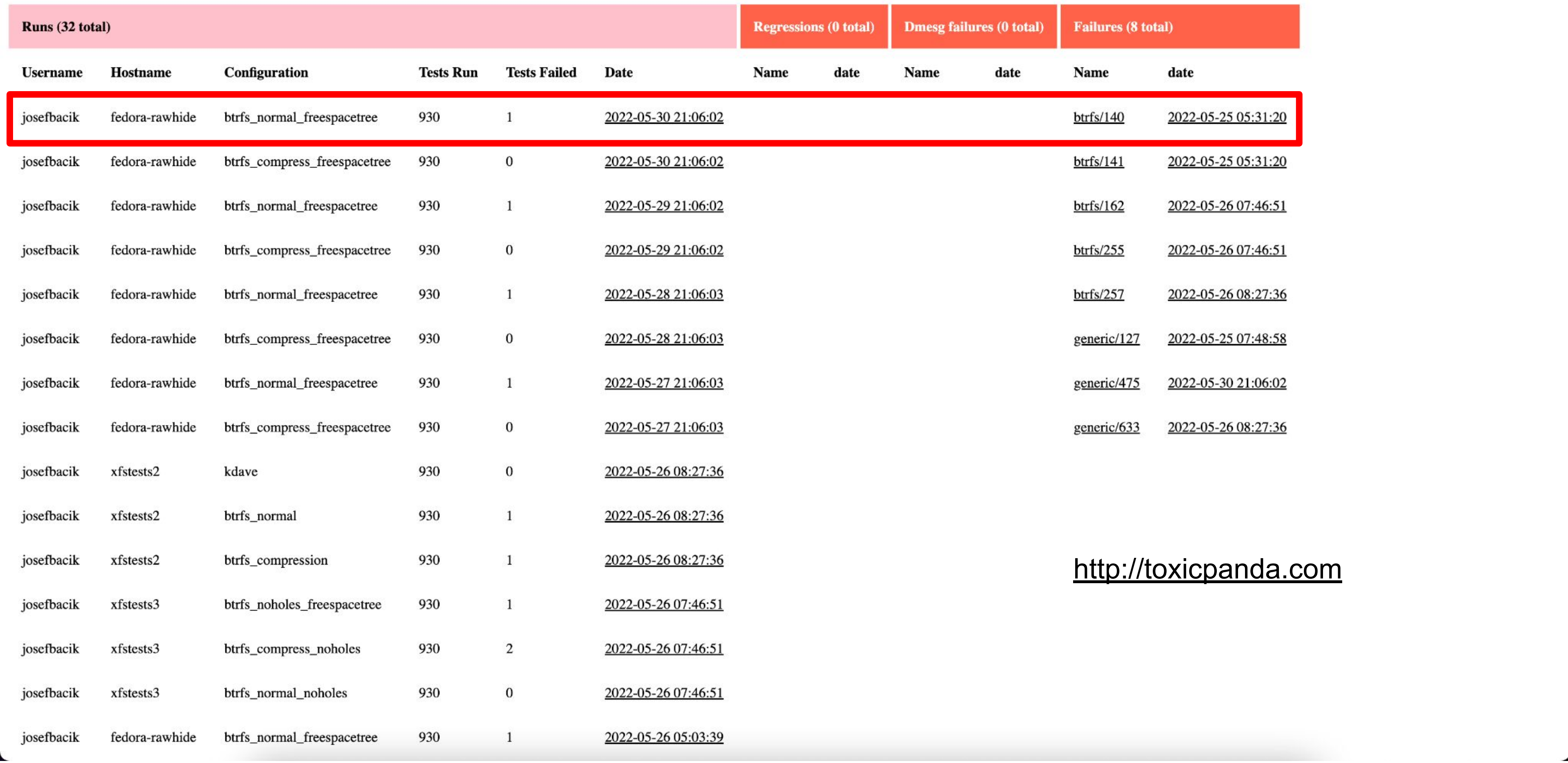
Runs (32 total)					
Username	**Hostname**	**Configuration**	**Tests Run**	**Tests Failed**	**Date**
josefbacik	fedora-rawhide	btrfs_normal_freespacetree	930	1	2022-05-30 21:06:02
josefbacik	fedora-rawhide	btrfs_compress_freespacetree	930	0	2022-05-30 21:06:02
josefbacik	fedora-rawhide	btrfs_normal_freespacetree	930	1	2022-05-29 21:06:02
josefbacik	fedora-rawhide	btrfs_compress_freespacetree	930	0	2022-05-29 21:06:02
josefbacik	fedora-rawhide	btrfs_normal_freespacetree	930	1	2022-05-28 21:06:03
josefbacik	fedora-rawhide	btrfs_compress_freespacetree	930	0	2022-05-28 21:06:03
josefbacik	fedora-rawhide	btrfs_normal_freespacetree	930	1	2022-05-27 21:06:03
josefbacik	fedora-rawhide	btrfs_compress_freespacetree	930	0	2022-05-27 21:06:03
josefbacik	xfstests2	kdave	930	0	2022-05-26 08:27:36
josefbacik	xfstests2	btrfs_normal	930	1	2022-05-26 08:27:36
josefbacik	xfstests2	btrfs_compression	930	1	2022-05-26 08:27:36
josefbacik	xfstests3	btrfs_noholes_freespacetree	930	1	2022-05-26 07:46:51
josefbacik	xfstests3	btrfs_compress_noholes	930	2	2022-05-26 07:46:51
josefbacik	xfstests3	btrfs_normal_noholes	930	0	2022-05-26 07:46:51
josefbacik	fedora-rawhide	btrfs_normal_freespacetree	930	1	2022-05-26 05:03:39

Regressions (0 total)	
Name	**date**

Dmesg failures (0 total)	
Name	**date**

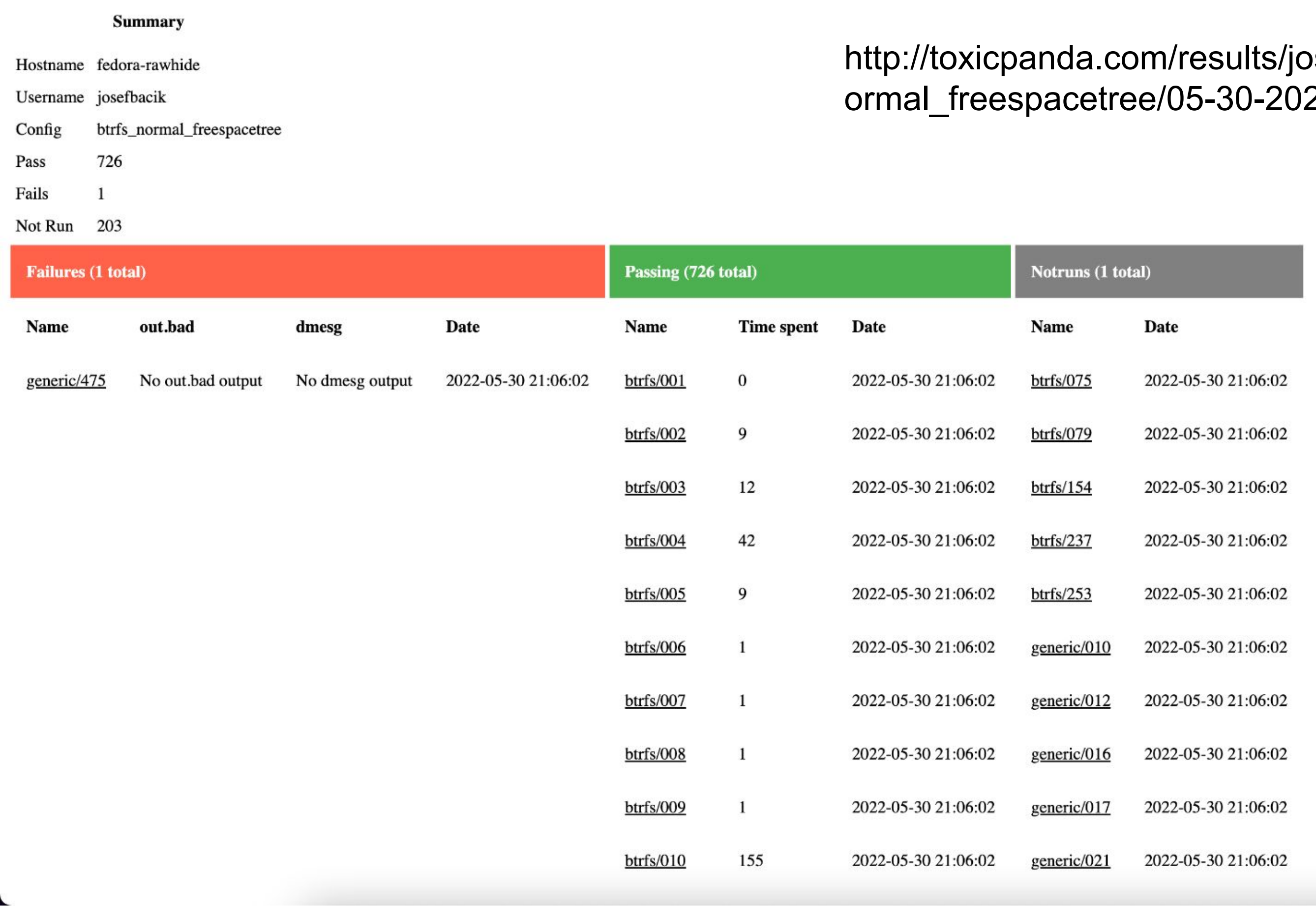
Failures (8 total)	
Name	**date**
btrfs/140	2022-05-25 05:31:20
btrfs/141	2022-05-25 05:31:20
btrfs/162	2022-05-26 07:46:51
btrfs/255	2022-05-26 07:46:51
btrfs/257	2022-05-26 08:27:36
generic/127	2022-05-25 07:48:58
generic/475	2022-05-30 21:06:02
generic/633	2022-05-26 08:27:36

http://toxicpanda.com



Runs (32 total)						Regressions (0 total)		Dmesg failures (0 total)		Failures (8 total)	
Username	Hostname	Configuration	Tests Run	Tests Failed	Date	Name	date	Name	date	Name	date
josefbacik	fedora-rawhide	btrfs_normal_freespacetree	930	1	2022-05-30 21:06:02					btrfs/140	2022-05-25 05:31:20
josefbacik	fedora-rawhide	btrfs_compress_freespacetree	930	0	2022-05-30 21:06:02					btrfs/141	2022-05-25 05:31:20
josefbacik	fedora-rawhide	btrfs_normal_freespacetree	930	1	2022-05-29 21:06:02					btrfs/162	2022-05-26 07:46:51
josefbacik	fedora-rawhide	btrfs_compress_freespacetree	930	0	2022-05-29 21:06:02					btrfs/255	2022-05-26 07:46:51
josefbacik	fedora-rawhide	btrfs_normal_freespacetree	930	1	2022-05-28 21:06:03					btrfs/257	2022-05-26 08:27:36
josefbacik	fedora-rawhide	btrfs_compress_freespacetree	930	0	2022-05-28 21:06:03					generic/127	2022-05-25 07:48:58
josefbacik	fedora-rawhide	btrfs_normal_freespacetree	930	1	2022-05-27 21:06:03					generic/475	2022-05-30 21:06:02
josefbacik	fedora-rawhide	btrfs_compress_freespacetree	930	0	2022-05-27 21:06:03					generic/633	2022-05-26 08:27:36
josefbacik	xfstests2	kdave	930	0	2022-05-26 08:27:36						
josefbacik	xfstests2	btrfs_normal	930	1	2022-05-26 08:27:36						
josefbacik	xfstests2	btrfs_compression	930	1	2022-05-26 08:27:36						
josefbacik	xfstests3	btrfs_noholes_freespacetree	930	1	2022-05-26 07:46:51						
josefbacik	xfstests3	btrfs_compress_noholes	930	2	2022-05-26 07:46:51						
josefbacik	xfstests3	btrfs_normal_noholes	930	0	2022-05-26 07:46:51						
josefbacik	fedora-rawhide	btrfs_normal_freespacetree	930	1	2022-05-26 05:03:39						

http://toxicpanda.com



Summary

Hostname	fedora-rawhide
Username	josefbacik
Config	btrfs_normal_freespacetree
Pass	726
Fails	1
Not Run	203

Failures (1 total)

Name	out.bad	dmesg	Date
generic/475	No out.bad output	No dmesg output	2022-05-30 21:06:02

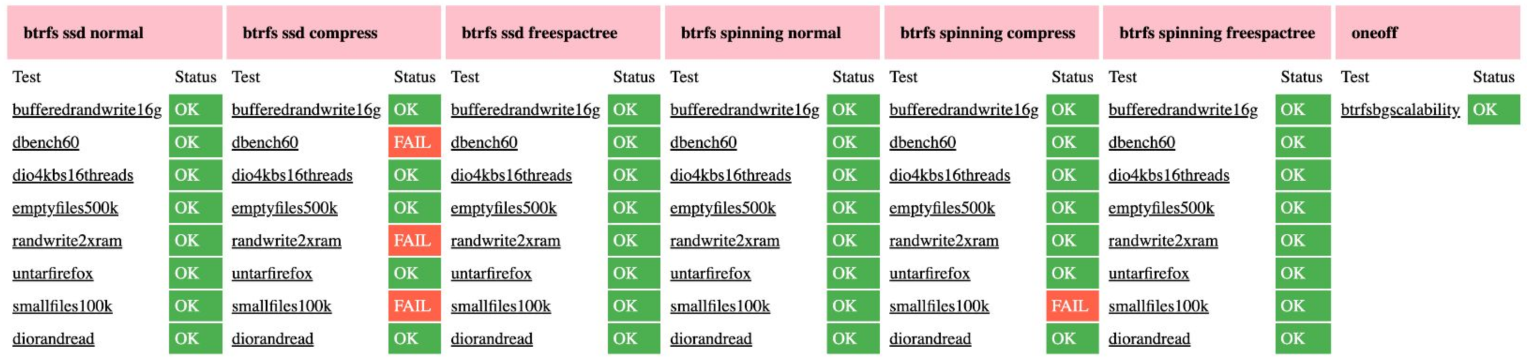
Passing (726 total)

Name	Time spent	Date
btrfs/001	0	2022-05-30 21:06:02
btrfs/002	9	2022-05-30 21:06:02
btrfs/003	12	2022-05-30 21:06:02
btrfs/004	42	2022-05-30 21:06:02
btrfs/005	9	2022-05-30 21:06:02
btrfs/006	1	2022-05-30 21:06:02
btrfs/007	1	2022-05-30 21:06:02
btrfs/008	1	2022-05-30 21:06:02
btrfs/009	1	2022-05-30 21:06:02
btrfs/010	155	2022-05-30 21:06:02

Notruns (1 total)

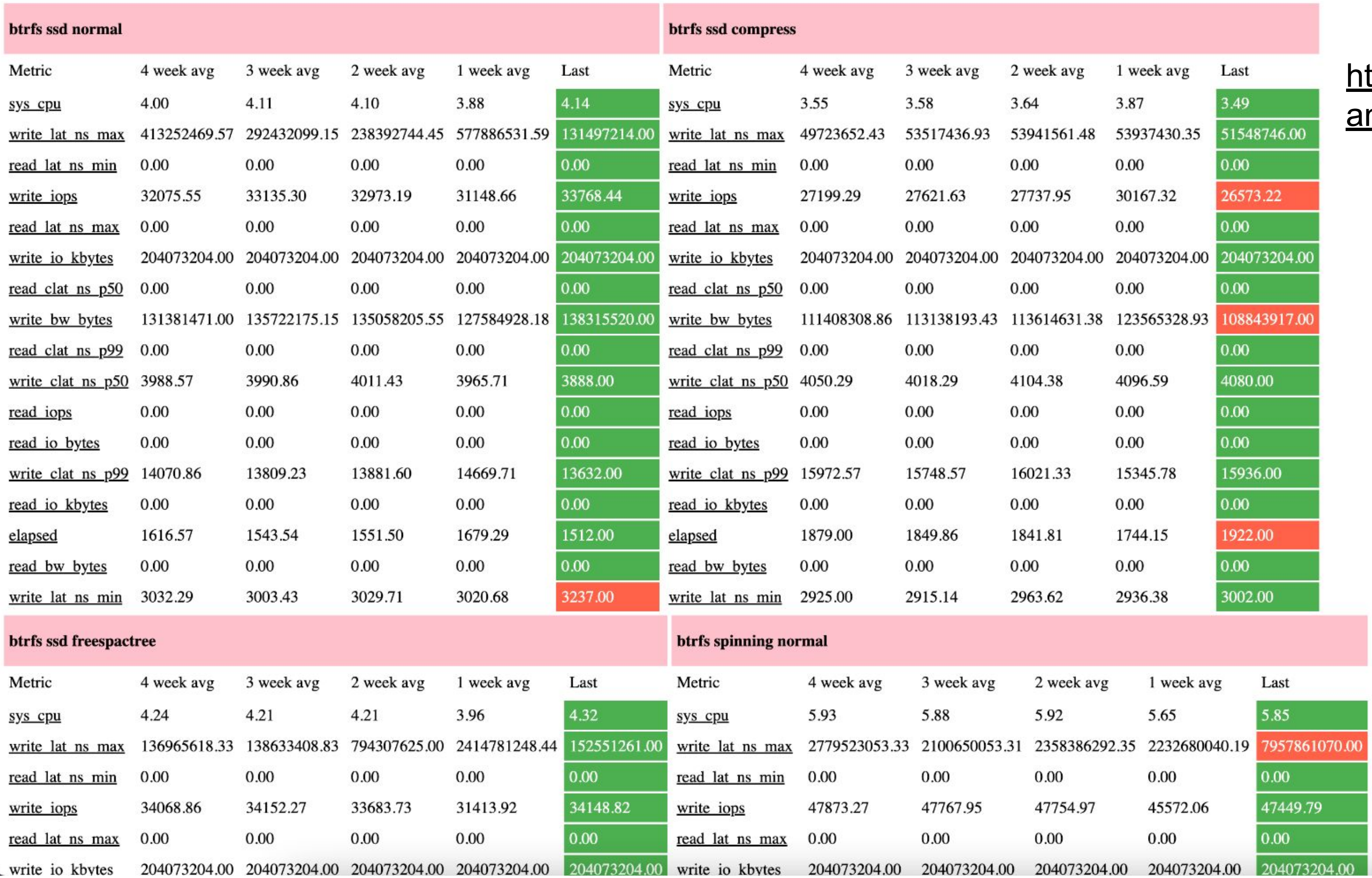
Name	Date
btrfs/075	2022-05-30 21:06:02
btrfs/079	2022-05-30 21:06:02
btrfs/154	2022-05-30 21:06:02
btrfs/237	2022-05-30 21:06:02
btrfs/253	2022-05-30 21:06:02
generic/010	2022-05-30 21:06:02
generic/012	2022-05-30 21:06:02
generic/016	2022-05-30 21:06:02
generic/017	2022-05-30 21:06:02
generic/021	2022-05-30 21:06:02

http://toxicpanda.com/results/josefbacik/fedora-rawhide/btrfs_normal_freespacetree/05-30-2022-21:06:02/index.html



btrfs ssd normal		btrfs ssd compress		btrfs ssd freespactree		btrfs spinning normal		btrfs spinning compress		btrfs spinning freespactree		oneoff	
Test	Status	Test	Status	Test	Status	Test	Status	Test	Status	Test	Status	Test	Status
bufferedrandwrite16g	OK	bufferedrandwrite16g	OK	bufferedrandwrite16g	OK	bufferedrandwrite16g	OK	bufferedrandwrite16g	OK	bufferedrandwrite16g	OK	btrfsbgscalability	OK
dbench60	OK	dbench60	FAIL	dbench60	OK	dbench60	OK	dbench60	OK	dbench60	OK		
dio4kbs16threads	OK	dio4kbs16threads	OK	dio4kbs16threads	OK	dio4kbs16threads	OK	dio4kbs16threads	OK	dio4kbs16threads	OK		
emptyfiles500k	OK	emptyfiles500k	OK	emptyfiles500k	OK	emptyfiles500k	OK	emptyfiles500k	OK	emptyfiles500k	OK		
randwrite2xram	OK	randwrite2xram	FAIL	randwrite2xram	OK	randwrite2xram	OK	randwrite2xram	OK	randwrite2xram	OK		
untarfirefox	OK	untarfirefox	OK	untarfirefox	OK	untarfirefox	OK	untarfirefox	OK	untarfirefox	OK		
smallfiles100k	OK	smallfiles100k	FAIL	smallfiles100k	OK	smallfiles100k	OK	smallfiles100k	FAIL	smallfiles100k	OK		
diorandread	OK	diorandread	OK	diorandread	OK	diorandread	OK	diorandread	OK	diorandread	OK		

http://toxicpanda.com/performance/



btrfs ssd normal

Metric	4 week avg	3 week avg	2 week avg	1 week avg	Last
sys_cpu	4.00	4.11	4.10	3.88	4.14
write_lat_ns_max	413252469.57	292432099.15	238392744.45	577886531.59	131497214.00
read_lat_ns_min	0.00	0.00	0.00	0.00	0.00
write_iops	32075.55	33135.30	32973.19	31148.66	33768.44
read_lat_ns_max	0.00	0.00	0.00	0.00	0.00
write_io_kbytes	204073204.00	204073204.00	204073204.00	204073204.00	204073204.00
read_clat_ns_p50	0.00	0.00	0.00	0.00	0.00
write_bw_bytes	131381471.00	135722175.15	135058205.55	127584928.18	138315520.00
read_clat_ns_p99	0.00	0.00	0.00	0.00	0.00
write_clat_ns_p50	3988.57	3990.86	4011.43	3965.71	3888.00
read_iops	0.00	0.00	0.00	0.00	0.00
read_io_bytes	0.00	0.00	0.00	0.00	0.00
write_clat_ns_p99	14070.86	13809.23	13881.60	14669.71	13632.00
read_io_kbytes	0.00	0.00	0.00	0.00	0.00
elapsed	1616.57	1543.54	1551.50	1679.29	1512.00
read_bw_bytes	0.00	0.00	0.00	0.00	0.00
write_lat_ns_min	3032.29	3003.43	3029.71	3020.68	3237.00

btrfs ssd compress

Metric	4 week avg	3 week avg	2 week avg	1 week avg	Last
sys_cpu	3.55	3.58	3.64	3.87	3.49
write_lat_ns_max	49723652.43	53517436.93	53941561.48	53937430.35	51548746.00
read_lat_ns_min	0.00	0.00	0.00	0.00	0.00
write_iops	27199.29	27621.63	27737.95	30167.32	26573.22
read_lat_ns_max	0.00	0.00	0.00	0.00	0.00
write_io_kbytes	204073204.00	204073204.00	204073204.00	204073204.00	204073204.00
read_clat_ns_p50	0.00	0.00	0.00	0.00	0.00
write_bw_bytes	111408308.86	113138193.43	113614631.38	123565328.93	108843917.00
read_clat_ns_p99	0.00	0.00	0.00	0.00	0.00
write_clat_ns_p50	4050.29	4018.29	4104.38	4096.59	4080.00
read_iops	0.00	0.00	0.00	0.00	0.00
read_io_bytes	0.00	0.00	0.00	0.00	0.00
write_clat_ns_p99	15972.57	15748.57	16021.33	15345.78	15936.00
read_io_kbytes	0.00	0.00	0.00	0.00	0.00
elapsed	1879.00	1849.86	1841.81	1744.15	1922.00
read_bw_bytes	0.00	0.00	0.00	0.00	0.00
write_lat_ns_min	2925.00	2915.14	2963.62	2936.38	3002.00

btrfs ssd freespactree

Metric	4 week avg	3 week avg	2 week avg	1 week avg	Last
sys_cpu	4.24	4.21	4.21	3.96	4.32
write_lat_ns_max	136965618.33	138633408.83	794307625.00	2414781248.44	152551261.00
read_lat_ns_min	0.00	0.00	0.00	0.00	0.00
write_iops	34068.86	34152.27	33683.73	31413.92	34148.82
read_lat_ns_max	0.00	0.00	0.00	0.00	0.00
write_io_kbytes	204073204.00	204073204.00	204073204.00	204073204.00	204073204.00

btrfs spinning normal

Metric	4 week avg	3 week avg	2 week avg	1 week avg	Last
sys_cpu	5.93	5.88	5.92	5.65	5.85
write_lat_ns_max	2779523053.33	2100650053.31	2358386292.35	2232680040.19	7957861070.00
read_lat_ns_min	0.00	0.00	0.00	0.00	0.00
write_iops	47873.27	47767.95	47754.97	45572.06	47449.79
read_lat_ns_max	0.00	0.00	0.00	0.00	0.00
write_io_kbytes	204073204.00	204073204.00	204073204.00	204073204.00	204073204.00

http://toxicpanda.com/performance/smallfiles100k.html

Independent solutions

Pros

- Tailored directly to the need of the subsystem
- Inspires test and benchmark writing

Cons

- No cross architecture, cross-config, etc coverage provided by framework.
- Maintainers need to spend a lot of their time getting something like this set up