Checking your work: Linux kernel testing and CI

Scaling reliability across the global upstream community

David Vernet
void@manifault.com

Kernel Recipes 2022 – Paris, France

Meta

Agenda

01 Disclaimers

02 How kernel tests are written

03 How kernel tests are run

04 What can we improve?

05 Q & A

06 Bonus: how to write a kselftest

01 Disclaimers

01 Disclaimers

1. I may be missing details of tools I’m not aware of

2. Presentation was crafted in the middle of the night over the Atlantic

02 How kernel tests are written

Pick your poison, there are a number of options

- kselftests (https://docs.kernel.org/dev-tools/kselftest.html)
- KUnit (https://docs.kernel.org/dev-tools/kunit/index.html)
- xfstests (https://git.kernel.org/pub/scm/fs/xfs/xfstests-dev.git/)
- Benchmarks (LKP @ https://github.com/intel/lkp-tests, Phoronix @ https://openbenchmarking.org/tests/pts)
- Fuzzers (https://github.com/google/syzkaller)
- Sanitizers (KASAN, kmemleak, …)
- Linux Test Project (https://github.com/linux-test-project/ltp)
- …

What are kselftests?

Testcases are instances of userspace programs

Commonly written in C, but need only be an executable file

Located in tree at `tools/testing/selftests`

File: **tools/testing/selftests/livepatch/test-callbacks.sh**
Size: **23.2 KB**

```
#!/bin/bash
# SPDX-License-Identifier: GPL-2.0
# Copyright (C) 2018 Joe Lawrence <joe.lawrence@redhat.com>

. $(dirname $0)/functions.sh

MOD_LIVEPATCH=test_klp_callbacks_demo
MOD_LIVEPATCH2=test_klp_callbacks_demo2
MOD_TARGET=test_klp_callbacks_mod
MOD_TARGET_BUSY=test_klp_callbacks_busy

setup_config


# Test a combination of loading a kernel module and a livepatch that
# patches a function in the first module.  Load the target module
# before the livepatch module.  Unload them in the same order.
#
# - On livepatch enable, before the livepatch transition starts,
#   pre-patch callbacks are executed for vmlinux and $MOD_TARGET (those
#   klp_objects currently loaded).  After klp_objects are patched
#   according to the klp_patch, their post-patch callbacks run and the
#   transition completes.
#
# - Similarly, on livepatch disable, pre-patch callbacks run before the
#   unpatching transition starts.  klp_objects are reverted, post-patch
#   callbacks execute and the transition completes.

start_test "target module before livepatch"

load_mod $MOD_TARGET
load_lp $MOD_LIVEPATCH
disable_lp $MOD_LIVEPATCH
unload_lp $MOD_LIVEPATCH
unload_mod $MOD_TARGET

check_result "% modprobe $MOD_TARGET
$MOD_TARGET: ${MOD_TARGET}_init
% modprobe $MOD_LIVEPATCH
livepatch: enabling patch '$MOD_LIVEPATCH'
livepatch: '$MOD_LIVEPATCH': initializing patching transition
$MOD_LIVEPATCH: pre_patch_callback: vmlinux
$MOD_LIVEPATCH: pre_patch_callback: $MOD_TARGET -> [MODULE_STATE_LIVE] Normal state
```

```
# cd tools/testing/selftests/kselftest_install/
# ls
kselftest  kselftest-list.txt  livepatch  run_kselftest.sh
# cat kselftest-list.txt
livepatch:test-livepatch.sh
livepatch:test-callbacks.sh
livepatch:test-shadow-vars.sh
livepatch:test-state.sh
livepatch:test-ftrace.sh
# ./run_kselftest.sh
TAP version 13
1..5
# selftests: livepatch: test-livepatch.sh
# TEST: basic function patching ... ok
# TEST: multiple livepatches ... ok
# TEST: atomic replace livepatch ... ok
ok 1 selftests: livepatch: test-livepatch.sh
# selftests: livepatch: test-callbacks.sh
# TEST: target module before livepatch ... ok
# TEST: module_coming notifier ... ok
# TEST: module_going notifier ... ok
# TEST: module_coming and module_going notifiers ... ok
# TEST: target module not present ... ok
# TEST: pre-patch callback -ENODEV ... ok
# TEST: module_coming + pre-patch callback -ENODEV ... ok
# TEST: multiple target modules ... ok
# TEST: busy target module ... ok
# TEST: multiple livepatches ... ok
# TEST: atomic replace ... ok
ok 2 selftests: livepatch: test-callbacks.sh
# selftests: livepatch: test-shadow-vars.sh
# TEST: basic shadow variable API ... ok
ok 3 selftests: livepatch: test-shadow-vars.sh
# selftests: livepatch: test-state.sh
# TEST: system state modification ... ok
# TEST: taking over system state modification ... ok
# TEST: compatible cumulative livepatches ... ok
# TEST: incompatible cumulative livepatches ... ok
ok 4 selftests: livepatch: test-state.sh
# selftests: livepatch: test-ftrace.sh
# TEST: livepatch interaction with ftrace_enabled sysctl ... ok
ok 5 selftests: livepatch: test-ftrace.sh
```

What are KUnit tests?

Unit testing framework for testing individual Linux kernel functions

Compiled into the kernel by specifying kconfig options

Testcases link directly against kernel symbols and kunit APIs, which are used to make assertions about expected return values of the kernel symbols

File: **drivers/clk/clk-gate_test.c**
Size: **13.2 KB**

```c
// SPDX-License-Identifier: GPL-2.0
/*
 * Kunit test for clk gate basic type
 */
#include <linux/clk.h>
#include <linux/clk-provider.h>
#include <linux/platform_device.h>

#include <kunit/test.h>

static void clk_gate_register_test_dev(struct kunit *test)
{
    struct clk_hw *ret;
    struct platform_device *pdev;

    pdev = platform_device_register_simple("test_gate_device", -1, NULL, 0);
    KUNIT_ASSERT_NOT_ERR_OR_NULL(test, pdev);

    ret = clk_hw_register_gate(&pdev->dev, "test_gate", NULL, 0, NULL,
                   0, 0, NULL);
    KUNIT_ASSERT_NOT_ERR_OR_NULL(test, ret);
    KUNIT_EXPECT_STREQ(test, "test_gate", clk_hw_get_name(ret));
    KUNIT_EXPECT_EQ(test, 0UL, clk_hw_get_flags(ret));

    clk_hw_unregister_gate(ret);
    platform_device_put(pdev);
}

static void clk_gate_register_test_parent_names(struct kunit *test)
{
    struct clk_hw *parent;
    struct clk_hw *ret;

    parent = clk_hw_register_fixed_rate(NULL, "test_parent", NULL, 0,
                        1000000);
    KUNIT_ASSERT_NOT_ERR_OR_NULL(test, parent);

    ret = clk_hw_register_gate(NULL, "test_gate", "test_parent", 0, NULL,
                   0, 0, NULL);
    KUNIT_ASSERT_NOT_ERR_OR_NULL(test, ret);
    KUNIT_EXPECT_PTR_EQ(test, parent, clk_hw_get_parent(ret));

    clk_hw_unregister_gate(ret);
```

```
$ ./tools/testing/kunit/kunit.py run
[23:20:29] Configuring KUnit Kernel ...
[23:20:29] Building KUnit Kernel ...
Populating config with:
$ make ARCH=um olddefconfig O=.kunit
Building with:
$ make ARCH=um --jobs=36 O=.kunit
In file included from ../arch/um/include/asm/processor-generic.h:13,
                 from ../arch/x86/um/asm/processor.h:41,
                 from ../include/linux/rcupdate.h:30,
                 from ../include/linux/rculist.h:11,
                 from ../include/linux/pid.h:5,
                 from ../include/linux/sched.h:14,
                 from ../include/linux/ptrace.h:6,
                 from ../arch/um/kernel/skas/syscall.c:7:
../arch/um/kernel/skas/syscall.c: In function ‘handle_syscall’:
../arch/x86/um/shared/sysdep/syscalls_64.h:18:4: warning: cast between incompati
ble function types from ‘long int (*)(void)’ to ‘long int (*)(long int,  long in
t,  long int,  long int,  long int,  long int)’ [-Wcast-function-type]         (
  (((long (*)(long, long, long, long, long, long)) \
    ^
../arch/x86/um/asm/ptrace.h:36:62: note: in definition of macro ‘PT_REGS_SET_SYS
CALL_RETURN’
 #define PT_REGS_SET_SYSCALL_RETURN(r, res) (PT_REGS_AX(r) = (res))
                                                              ^~~
../arch/um/kernel/skas/syscall.c:46:5: note: in expansion of macro ‘EXECUTE_SYSC
ALL’
     EXECUTE_SYSCALL(syscall, regs));
     ^~~~~~~~~~~~~~

[23:20:56] Starting KUnit Kernel (1/1)...
[23:20:56] ============================================================
[23:20:57] ================ time_test_cases (1 subtest) ================
[23:20:57] [PASSED] time64_to_tm_test_date_range
[23:20:57] ================== [PASSED] time_test_cases ==================
[23:20:57] ================ sysctl_test (10 subtests) =================
[23:20:57] [PASSED] sysctl_test_api_dointvec_null_tbl_data
[23:20:57] [PASSED] sysctl_test_api_dointvec_table_maxlen_unset
[23:20:57] [PASSED] sysctl_test_api_dointvec_table_len_is_zero
[23:20:57] [PASSED] sysctl_test_api_dointvec_table_read_but_position_set
[23:20:57] [PASSED] sysctl_test_dointvec_read_happy_single_positive
[23:20:57] [PASSED] sysctl_test_dointvec_read_happy_single_negative
[23:20:57] [PASSED] sysctl_test_dointvec_write_happy_single_positive
[23:20:57] [PASSED] sysctl_test_dointvec_write_happy_single_negative
[23:20:57] [PASSED] sysctl_test_api_dointvec_write_single_less_int_min
```

```
[23:20:57] [SKIPPED] test_small_hole_assigned_static_all
[23:20:57] [SKIPPED] test_big_hole_assigned_static_all
[23:20:57] [SKIPPED] test_trailing_hole_assigned_static_all
[23:20:57] [PASSED] test_packed_assigned_static_all
[23:20:57] [SKIPPED] test_small_hole_assigned_dynamic_all
[23:20:57] [SKIPPED] test_big_hole_assigned_dynamic_all
[23:20:57] [SKIPPED] test_trailing_hole_assigned_dynamic_all
[23:20:57] [PASSED] test_packed_assigned_dynamic_all
[23:20:57] [SKIPPED] test_small_hole_assigned_copy
[23:20:57] [SKIPPED] test_big_hole_assigned_copy
[23:20:57] [SKIPPED] test_trailing_hole_assigned_copy
[23:20:57] [PASSED] test_packed_assigned_copy
[23:20:57] [SKIPPED] test_u8_none
[23:20:57] [SKIPPED] test_u16_none
[23:20:57] [SKIPPED] test_u32_none
[23:20:57] [SKIPPED] test_u64_none
[23:20:57] [SKIPPED] test_char_array_none
[23:20:57] [SKIPPED] test_switch_1_none
[23:20:57] [SKIPPED] test_switch_2_noner
[23:20:57] [SKIPPED] test_small_hole_none
[23:20:57] [SKIPPED] test_big_hole_none0
[23:20:57] [SKIPPED] test_trailing_hole_none
[23:20:57] [SKIPPED] test_packed_none
[23:20:57] [SKIPPED] test_user
[23:20:57] ==================== [PASSED] stackinit ====================
[23:20:57] =============== qos-kunit-test (3 subtests) ===============
[23:20:57] [PASSED] freq_qos_test_min
[23:20:57] [PASSED] freq_qos_test_maxdef
[23:20:57] [PASSED] freq_qos_test_readd
[23:20:57] ================= [PASSED] qos-kunit-test =================
[23:20:57] =============== property-entry (7 subtests) ===============
[23:20:57] [PASSED] pe_test_uints
[23:20:57] [PASSED] pe_test_uint_arrays
[23:20:57] [PASSED] pe_test_strings
[23:20:57] [PASSED] pe_test_bool
[23:20:57] [PASSED] pe_test_move_inline_u8
[23:20:57] [PASSED] pe_test_move_inline_str
[23:20:57] [PASSED] pe_test_reference
[23:20:57] ================= [PASSED] property-entry =================
[23:20:57] ============================================================
[23:20:57] Testing complete. Passed: 137, Failed: 0, Crashed: 0, Skipped: 36, Er
rors: 0
[23:20:57] Elapsed time: 27.645s total, 0.001s configuring, 26.573s building, 1.
044s running
```

What are xfstests?

Filesystem regression test suite (https://git.kernel.org/pub/scm/fs/xfs/xfstests-dev.git/)

Tests are categorized according to whether they’re global, shared between a subset of FSs, or specific to one FS

Tests use common logic for bootstrapping block devices, etc

Located in a separate repository

And more test repos housed in external repositories

Linux Kernel Performance (https://github.com/intel/lkp-tests)

Phoronix (https://openbenchmarking.org/tests/pts)

Linux Test Project (https://github.com/linux-test-project/ltp)

03 How kernel tests are run

Pick your poison, there are a few options

- KernelCI (https://foundation.kernelci.org)
- LKP / kernel test robot (https://01.org/lkp/documentation/0-day-brief-introduction)
- Patchwork + github + extra magic (https://patchwork.kernel.org/project/netdevbpf/list/)
- syzbot (https://syzkaller.appspot.com/upstream)
- Maintainers' private machines (e.g. Josef Bacik's btrfs dashboards: http://toxicpanda.com/)
- Thorsten Leemhuis' regzbot (https://linux-regtracking.leemhuis.info/regzbot/mainline/)

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

Home | Jobs | Builds | Tests | SoCs | Info

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

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	

Home | Jobs | Builds | Tests | SoCs | Info

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

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

Filter the results

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

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

Filter the results

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

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	⚠

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

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

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

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

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

Home | Jobs | Builds | Tests | SoCs | Info

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

Filter the results

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

index : kernel/git/torvalds/linux.git

master switch

Linux kernel source tree

Linus Torvalds

about summary refs log tree commit diff stats

log msg search

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
```

Home | Jobs | Builds | Tests | SoCs | Info

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	✓

Home | Jobs | Builds | Tests | SoCs | Info

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	✓

Home | Jobs | Builds | Tests | SoCs | Info

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

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

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-

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
Node name	build-j141520-mips-gcc-10-decstation-64-defconfig-zqq9f
Release	5.4.0-1065-azure
Full release	#68~18.04.1-Ubuntu SMP Fri Dec 3 14:08:44 UTC 2021

Machine type	x86_64
CPU	Intel(R) Xeon(R) Platinum 8272CL CPU @ 2.60GHz

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

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?)

Overview | Resources

Rapid Evolution of Linux Development

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

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

Rapid Evolution of Linux Development

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

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

Give Feedback

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

MI01.01.Org https://lists.01.org/hyperkitty/list/kbuild-all@lists.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...

2022

May
April
March
February
January

2021

2020

2019

List overview

Download

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]...

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>
CC: "Greg Kroah-Hartman" <gregkh(a)linuxfoundation.org>
CC: Srinivas Kandagatla <srinivas.kandagatla(a)linaro.org>

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>
Reported-by: Julia Lawall <julia.lawall(a)lip6.fr>

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>
CC: "Greg Kroah-Hartman" <gregkh(a)linuxfoundation.org>
CC: Srinivas Kandagatla <srinivas.kandagatla(a)linaro.org>
CC: Andy Gross <agross(a)kernel.org>
CC: Bjorn Andersson <bjorn.andersson(a)linaro.org>
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>

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

newer

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

older

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.

(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

17 days inactive

40 days old

lkp@lists.01.org

Manage subscription

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@

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

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/testcase/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?)

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

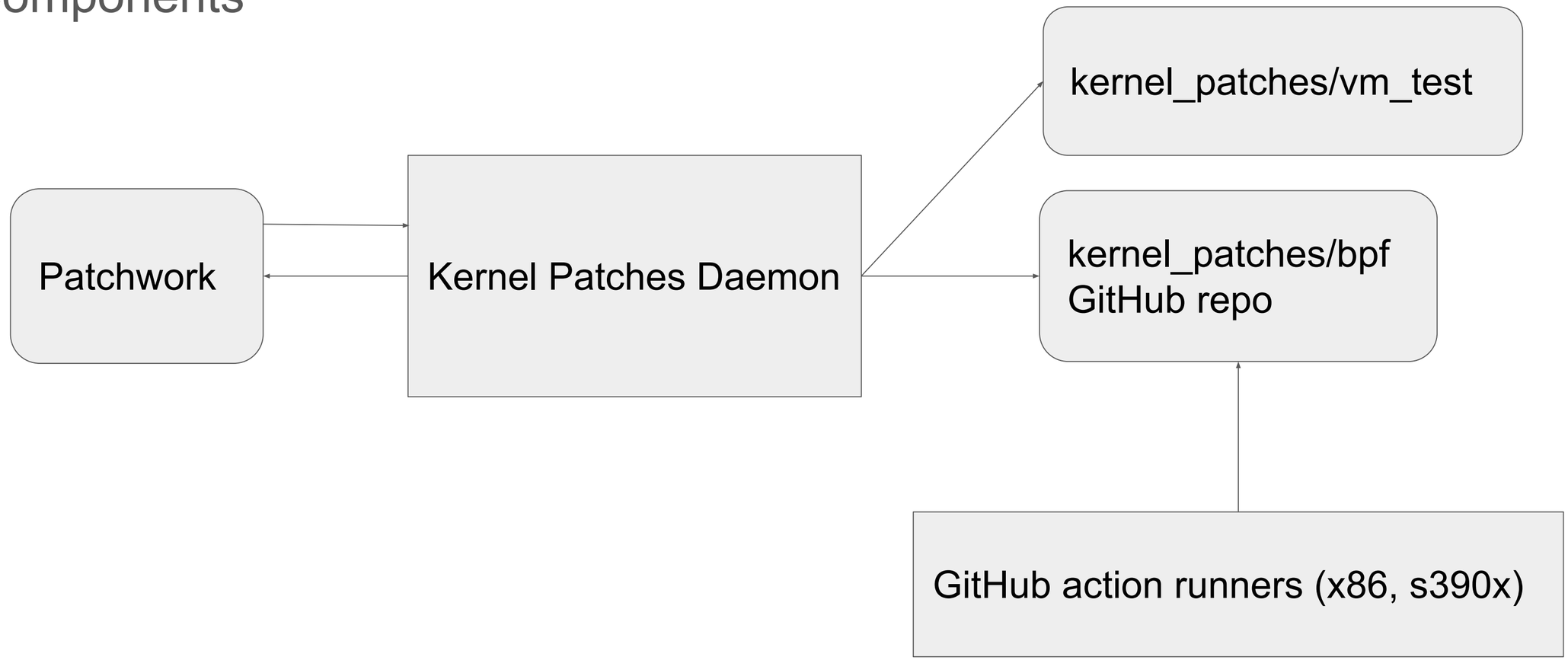
Patchwork Netdev + BPF | Patches | Bundles | About this project | Login | Register | Mail settings

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

Components



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)

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

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

[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
```

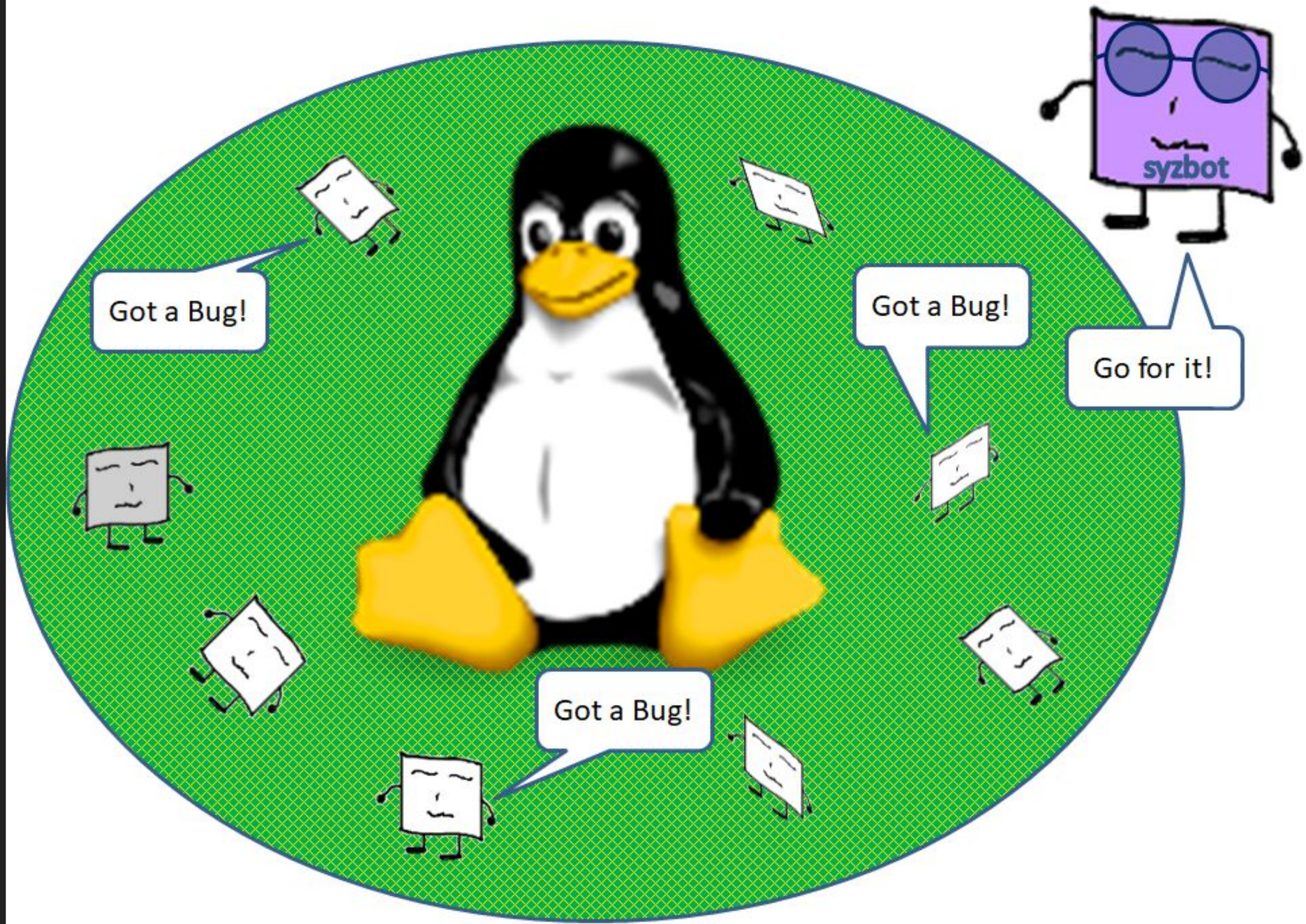
Patchwork

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)


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)						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

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

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

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

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

04 What can be improved?

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

04 What can be improved?

Let’s start by talking about CI

All of the CI systems we've covered have roughly the same, or at least similar, goals

Run tests on some matrix of configurations and architectures

When regressions are detected, provide signal:

- Ideally before patches are merged
- Otherwise, bisect and detect the bad patch automatically

All of the CI systems do a subset of things well

KernelCI has a great UI, gets a lot of test coverage and provides detailed information

LKP / kernel test robot / 0-day detects regressions for all patches sent to the list, and pings vger when a regression is detected. It also runs tests not included in the source tree, including benchmarks

Patchwork / BPF also has a great UI, makes it easy for developers to test locally, and provides signal for all patches sent to the BPF lists. The signal is also highly reliable, due to BPF selftests being deterministic and fast.

Can we combine forces?

As maintainers / kernel developers, for the purposes of testing the kernel, can we break anything out into shared code?

- Patch bisection
- Invoking kselftests, kunit, interpreting TAP output

04 What can be improved?

What about our approach to writing tests?

kselftests is great, but has room for improvement

Was originally intended as a dumping ground for tests that would often bit rot on individual developers' servers

```
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>
```

Allow for more comprehensive kselftest configurations

The maintainers of each test suite know best how it should be configured

Allow selftest suites to be configured to advertise:

- **State**: Stable, flaky, unstable
- **Support**: Supported architectures, unsupported config options (not just what’s necessary to run which is what exists today)
- Trees and branches to run on
- Frequency of runs + how to invoke test for each frequency

Add more tests!

Great way to test your newly added APIs (both design **and** correctness)

Leverage the excellent infrastructure being developed in tools like KernelCI

Add your tests to the tree

Out-of-tree tests

Nothing at all wrong with having them (in fact they provide a ton of value today), but…

Having tests which inform the "official" stability, performance, etc for the kernel, should probably reside in the kernel tree as a general rule

- Allows tests to be controlled and configured by maintainers

CI systems can always pull tests from multiple sources

04 What can be improved?

…and what do we need to avoid?

Annoying maintainers

Having a CI system should **alleviate** pressure on maintainers

Things can get tricky though

- Flaky tests
- Tests failing after merge

If tests waste people’s time, they are providing negative value

If CI systems spam upstream lists, they are providing negative value

Not all tests created equal

Need a high threshold (which we currently have) for when failing CI runs should email upstream lists

- Build regressions are a very stable and reliable signal
- If a testrun fails, it’s less clear. Could be flaky, broken test, failing hardware on the host, etc.

How failing tests are interpreted should be up to the maintainers of a subsystem

For subsystems like RCU and BPF, test failures are a strong signal, as tests are actively fixed if flakiness is observed

For subsystems like cgroup, it’s less clear. Some testcases (such as test_cpu.c and test_memcontrol.c) are validating heuristic behavior

05 Q & A


Meta

06 Bonus: How to write a kselftest

Anatomy of a kselftest suite – livepatch

```
$ pwd
/home/dvernet/upstream/livepatching
$ ls tools/testing/selftests/livepatch/
config  functions.sh  Makefile  README  settings  test-callbacks.sh  test-ftrace.sh  test-livepatch.sh
 test-shadow-vars.sh  test-state.sh
```

config file contains kconfig options required to build and run the suite

Makefile contains recipes for compiling testcases, and variables that are consumed by the kselftest build system

kselftests example – livepatch config file and Makefiles

```
$ bat tools/testing/selftests/livepatch/config
       | File: tools/testing/selftests/livepatch/config
       | Size: 66 B
-------+------------------------------------------------
   1   | CONFIG_LIVEPATCH=y
   2   | CONFIG_DYNAMIC_DEBUG=y
   3   | CONFIG_TEST_LIVEPATCH=m
```

```
$ bat tools/testing/selftests/livepatch/Makefile
       | File: tools/testing/selftests/livepatch/Makefile
       | Size: 229 B
-------+------------------------------------------------
   1   | # SPDX-License-Identifier: GPL-2.0
   2   |
   3   | TEST_PROGS_EXTENDED := functions.sh
   4   | TEST_PROGS := \
   5   |     test-livepatch.sh \
   6   |     test-callbacks.sh \
   7   |     test-shadow-vars.sh \
   8   |     test-state.sh \
   9   |     test-ftrace.sh
  10   |
  11   | TEST_FILES := settings
  12   |
  13   | include ../lib.mk
```

```
$ make -j -C tools/testing/selftests install TARGETS=livepatch
make: Entering directory '/data/users/dvernet/upstream/livepatching/tools/testing/selftests'
make --no-builtin-rules ARCH=x86 -C ../../.. headers_install
make[1]: Entering directory '/data/users/dvernet/upstream/livepatching'
Emit Tests for livepatch\nmake: Leaving directory '/data/users/dvernet/upstream/livepatching/tools/
testing/selftests'

<snip>

$
$ ls tools/testing/selftests/kselftest_install
kselftest  kselftest-list.txt  livepatch  run_kselftest.sh
$ cat tools/testing/selftests/kselftest_install/kselftest-list.txt
livepatch:test-livepatch.sh
livepatch:test-callbacks.sh
livepatch:test-shadow-vars.sh
livepatch:test-state.sh
livepatch:test-ftrace.sh
```

```
# cd tools/testing/selftests/kselftest_install/
# ls
kselftest  kselftest-list.txt  livepatch  run_kselftest.sh
# cat kselftest-list.txt
livepatch:test-livepatch.sh
livepatch:test-callbacks.sh
livepatch:test-shadow-vars.sh
livepatch:test-state.sh
livepatch:test-ftrace.sh
# ./run_kselftest.sh
TAP version 13
1..5
# selftests: livepatch: test-livepatch.sh
# TEST: basic function patching ... ok
# TEST: multiple livepatches ... ok
# TEST: atomic replace livepatch ... ok
ok 1 selftests: livepatch: test-livepatch.sh
# selftests: livepatch: test-callbacks.sh
# TEST: target module before livepatch ... ok
# TEST: module_coming notifier ... ok
# TEST: module_going notifier ... ok
# TEST: module_coming and module_going notifiers ... ok
# TEST: target module not present ... ok
# TEST: pre-patch callback -ENODEV ... ok
# TEST: module_coming + pre-patch callback -ENODEV ... ok
# TEST: multiple target modules ... ok
# TEST: busy target module ... ok
# TEST: multiple livepatches ... ok
# TEST: atomic replace ... ok
ok 2 selftests: livepatch: test-callbacks.sh
# selftests: livepatch: test-shadow-vars.sh
# TEST: basic shadow variable API ... ok
ok 3 selftests: livepatch: test-shadow-vars.sh
# selftests: livepatch: test-state.sh
# TEST: system state modification ... ok
# TEST: taking over system state modification ... ok
# TEST: compatible cumulative livepatches ... ok
# TEST: incompatible cumulative livepatches ... ok
ok 4 selftests: livepatch: test-state.sh
# selftests: livepatch: test-ftrace.sh
# TEST: livepatch interaction with ftrace_enabled sysctl ... ok
ok 5 selftests: livepatch: test-ftrace.sh
```


Meta