

Инструменты тестирования ядра Linux

Денис Ефремов <efremov@ispras.ru>
Алексей Хорошилов <khoroshilov@ispras.ru>



Отдельные тесты (модули/приложения)

- Могут быть найдены:
 - `grep --include='Kconfig*' \`
`-nre 'SELFTEST|BENCHMARK'`
- Нет общей структуры
 - Нет тестового протокола
 - Результаты не структурированы
 - Не/Зависят от наличия аппаратуры
- Тест начинается в момент загрузки
 - Во время инициализации ядра для встроенных
- Пользовательское пространство ([fault-injection](#), [nvdim](#), [radix-tree](#), [scatterlist](#), [vsock](#)) в [tools/testing](#)

```
kernel/dma/Kconfig:225:config DMA_MAP_BENCHMARK
kernel/trace/Kconfig:692:config TRACEPOINT_BENCHMARK
kernel/trace/Kconfig:722:config RING_BUFFER_BENCHMARK
kernel/trace/Kconfig:817:config FTRACE_SELFTEST
kernel/trace/Kconfig:823:      select FTRACE_SELFTEST
arch/xtensa/Kconfig.debug:24:config S32C1I_SELFTEST
arch/powerpc/Kconfig.debug:55:config CODE_PATCHING_SELFTEST
arch/powerpc/Kconfig.debug:77:config FTR_FIXUP_SELFTEST
arch/powerpc/Kconfig.debug:81:config MSI_BITMAP_SELFTEST
arch/Kconfig:74:config STATIC_KEYS_SELFTEST
arch/Kconfig:80:config STATIC_CALL_SELFTEST
arch/um/Kconfig:180:      depends on !RAID6_PQ_BENCHMARK
arch/s390/Kconfig:949:config S390_UNWIND_SELFTEST
arch/x86/Kconfig.debug:120:config X86_DECODER_SELFTEST
arch/x86/Kconfig.debug:183:config DEBUG_NMI_SELFTEST
arch/x86/Kconfig.debug:195:config DEBUG_IMR_SELFTEST
arch/arc/Kconfig:590:      default y if STATIC_KEYS_SELFTEST
lib/Kconfig:14:config RAID6_PQ_BENCHMARK
lib/Kconfig:148:config CRC32_SELFTEST
lib/Kconfig:263:config RANDOM32_SELFTEST
lib/Kconfig:519:config GLOB_SELFTEST
lib/Kconfig:672:config STRING_SELFTEST
lib/Kconfig.debug:591:config DEBUG_OBJECTS_SELFTEST
lib/Kconfig.debug:1393:config DEBUG_LOCKING_API_SELFTESTS
lib/Kconfig.debug:1418:config WW_MUTEX_SELFTEST
lib/Kconfig.debug:2090:config ATOMIC64_SELFTEST
lib/Kconfig.debug:2252:config FIND_BIT_BENCHMARK
lib/Kconfig.kcsan:64:config KCSAN_SELFTEST
```

KUnit (модули*)

- Фреймворк юнит тестов для ядра Linux
 - Разработан Brendan Higgings (Google)
 - Включен в ядро с версии [v5.5](#)
 - `./tools/testing/kunit/kunit.py run`
- Предназначен для небольших, изолированных тестов
 - Но не ограничен ими
- Целевая архитектура – [User-Mode Linux \(UML\)](#) (-KASAN)
 - Используется Android в [kernel networking tests](#)
- Особенности фреймворка:
 - Общая инфраструктура для юнит тестов
 - Объединение тестов в группы
 - Параметризация тестов
 - Мокирование интерфейсов
- Результаты ([TAP protocol](#)):
 - Лог ядра (dmesg)
 - `Debugfs /sys/kernel/debug/kunit/<test-suite>`
- `fgrep --include='Kconfig*' -nre KUNIT_TEST`
 - `KUNIT_ALL_TESTS`

```
[15:42:46] Configuring KUnit Kernel ...
Generating .config ...
[15:42:49] Building KUnit Kernel ...
[15:43:13] Starting KUnit Kernel ...
[15:43:15] =====
[15:43:15] ===== [PASSED] kunit-try-catch-test =
[15:43:15] [PASSED] kunit_test_try_catch_successful
[15:43:15] [PASSED] kunit_test_try_catch_unsuccessful
[15:43:15] =====
[15:43:15] ===== [PASSED] kunit-resource-test =
[15:43:15] [PASSED] kunit_resource_test_init_resource
[15:43:15] [PASSED] kunit_resource_test_alloc_resource
[15:43:15] [PASSED] kunit_resource_test_destroy_resource
[15:43:15] [PASSED] kunit_resource_test_cleanup_resource
[15:43:15] [PASSED] kunit_resource_test_proper_free_resource
[15:43:15] [PASSED] kunit_resource_test_static_resource
[15:43:15] [PASSED] kunit_resource_test_named_resource
```

КТФ (Kernel Test Framework) (гибридные)

- Фреймворк юнит тестов для ядра Linux от Oracle
 - [На текущий момент не включен в ядро](#)
- Netlink для kernel<->userspace взаимодействия
- Kprobes/Kretprobes для перехвата внутренних API
- Автоматизация доступа к [внутренним символам ядра](#)
- Особенности фреймворка :
 - Общая инфраструктура для юнит тестов
 - Объединение тестов в группы ([test suites](#))
 - Параметризация тестов ([fixtures](#), [contexts](#))
 - Мокирование интерфейсов
 - [Упрощенное покрытие](#)
 - [Гибридные тесты](#)
- Результаты (протокол TAP):
 - Через сокет netlink
 - Debugfs /sys/kernel/debug/ktf/results/
- Не существует публичного тестового набора

```
[=====] Running 1 test from 1 test suite.
[-----] Global test environment set-up.
[-----] 1 test from examples
[ RUN      ] examples.hello_fail
/home/work/workspace/ktf/examples/hello.c:21: Failure
Failure 'false' occurred
[  FAILED  ] examples.hello_fail, where GetParam() = "h
[-----] 1 test from examples (0 ms total)

[-----] Global test environment tear-down
[=====] 1 test from 1 test suite ran. (0 ms total)
[  PASSED  ] 0 tests.
[  FAILED  ] 1 test, listed below:
[  FAILED  ] examples.hello_fail, where GetParam() = "h
```

Kselftest

(приложения регрессионные/интеграционные тесты)

- [tools/testing/selftests](#) ([first in v3.16 release](#))
- Сборка->Установка->Загрузка->Запуск тестов
- Предназначены для базового тестирования
 - Менее чем за 20 min
 - `make kselftest`
- Новые модули включаются в ядра с тестами
 - [wireguard](#), [landlock](#), [SafeSetSID](#), [syscall user dispatch](#), [SGX](#)...
- LKFT (Linaro) [версии kernel<->tests](#):
 - linux-next использует selftest из linux-next
 - mainline использует selftest из mainline
 - Latest stable использует selftest из latest stable
 - *Остальные LTS используют selftest из latest stable*

```
ok 1 selftests: ptrace: get_syscall_info
# selftests: ptrace: peeksignfo
# PASS
ok 2 selftests: ptrace: peeksignfo
# selftests: ptrace: vmaccess
# TAP version 13
# 1..2
# # Starting 2 tests from 1 test cases.
# # RUN          global.vmassess ...
# #             OK global.vmassess
# ok 1 global.vmassess
# # RUN          global.attach ...
# # attach: Test terminated by timeout
# #             FAIL global.attach
# not ok 2 global.attach
# # FAILED: 1 / 2 tests passed.
# # Totals: pass:1 fail:1 xfail:0 xpass:0
not ok 3 selftests: ptrace: vmaccess # exit
make[2]: Leaving directory '/home/work/work'
make[2]: Entering directory '/home/work/work'
gcc -Wall -O2 -g -fsanitize=address -fsanitize=
est
```

Ktest (тестирование сборки/загрузки)

STARTING AUTOMATED TESTS

- Автоматизация сборки, установки, загрузки и тестирования
- В ядре с 2010 (2.6.38) (tools/testing/ktest)
 - Разрабатывается Steven Rostedt
- Тестирование сборки:
 - Randconfig (config bisecting)
 - Проверка патчей из листов рассылок, git
- Тестирование загрузки:
 - Проверяет лог ядра
 - Возможно выполнение тестов после
 - Bisecting и тестирование патчсетов
- Требования:
 - Хост и целевая машины (hardware/virtual)
 - SSH + root доступ к целевой машине
 - Удаленный контроль питания целевой машины

DEFAULT OPTIONS:

BISECT_MANUAL = 0

BISECT_SKIP = 1

BISECT_SLEEP_TIME = 60

BISECT_TRIES = 1

BOOTED_TIMEOUT = 1

BUILD_DIR = /home/work/workspa

BUILD_NOCLEAN = 0

BUILD_OPTIONS = -j12

BUILD_TARGET = arch/x86/boot/b

BUILD_TYPE = oldconfig

CLEAR_LOG = 0

CLOSE_CONSOLE_SIGNAL = INT

CONNECT_TIMEOUT = 25

DETECT_TRIPLE_FAULT = 1

DIE_ON_FAILURE = 1

EMAIL_ON_ERROR = 1

EMAIL_WHEN_CANCELED = 0

EMAIL_WHEN_FINISHED = 1

Известные тестовые наборы (приложения)

- [Linux Test Project \(LTP\)](#) – регрессионные, конформанс тесты. Поддерживается совместно многими компаниями. Используется всеми CI системами. [Пример отчёта](#)
- [Linux Kernel Performance tests \(LKP\)](#) – тесты виртуальной памяти, подсистемы I/O, планировщика процессов, файловых систем, сети, и др. Поддерживается Intel, используется 0-Day CI. [Пример отчёта](#)
- [xfstests](#) – набор регрессионных тестов для большинства файловых систем Linux
- [syzkaller-repros](#) – репродюсеры syzkaller
- [KVM unit tests](#) – небольшие гостевые ОС для тестов гипервизора
- ...

Вопросы?