YVR18-308: Kernel Bugs and Regressions Debugging Best Practices

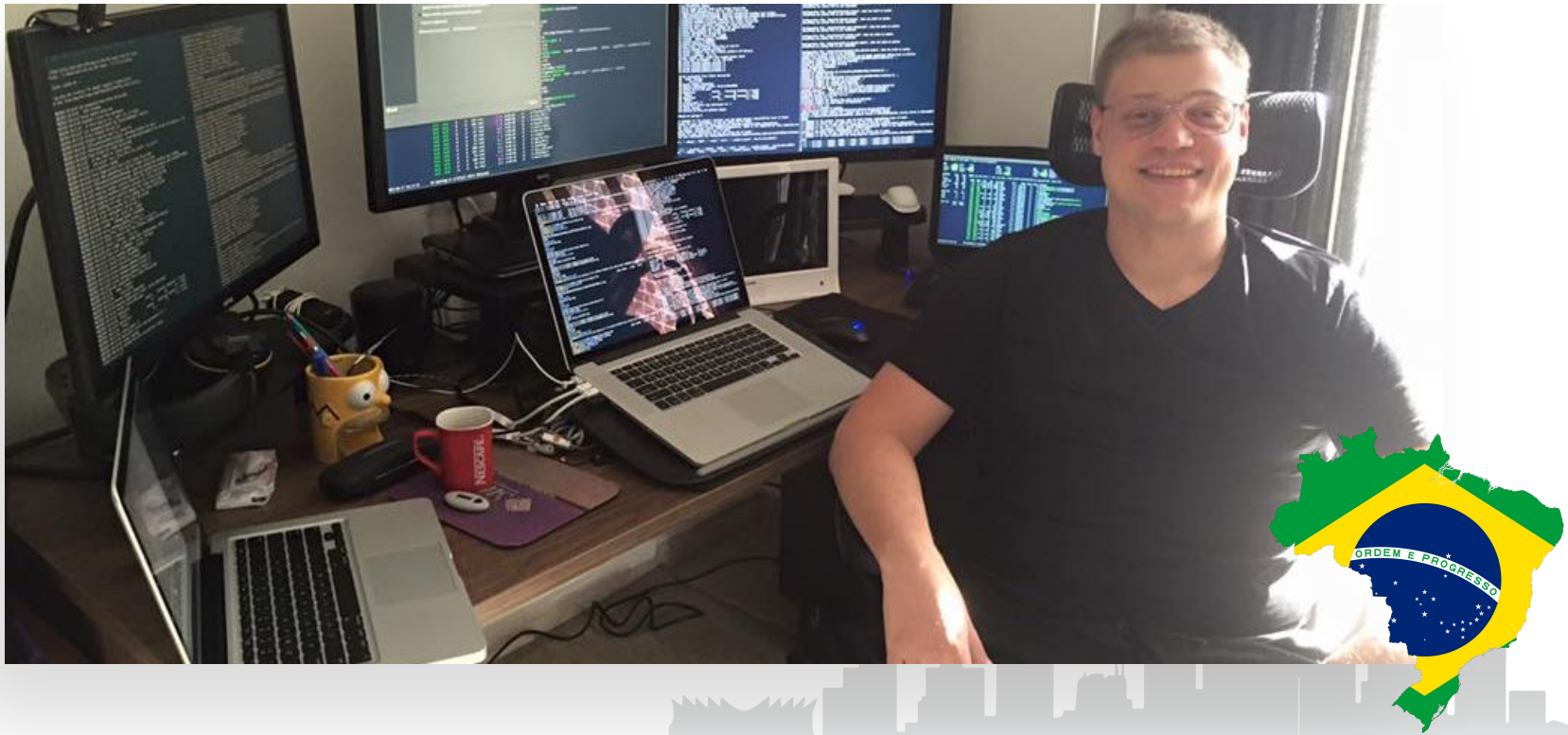
Rafael David Tinoco
Kernel Validation Team

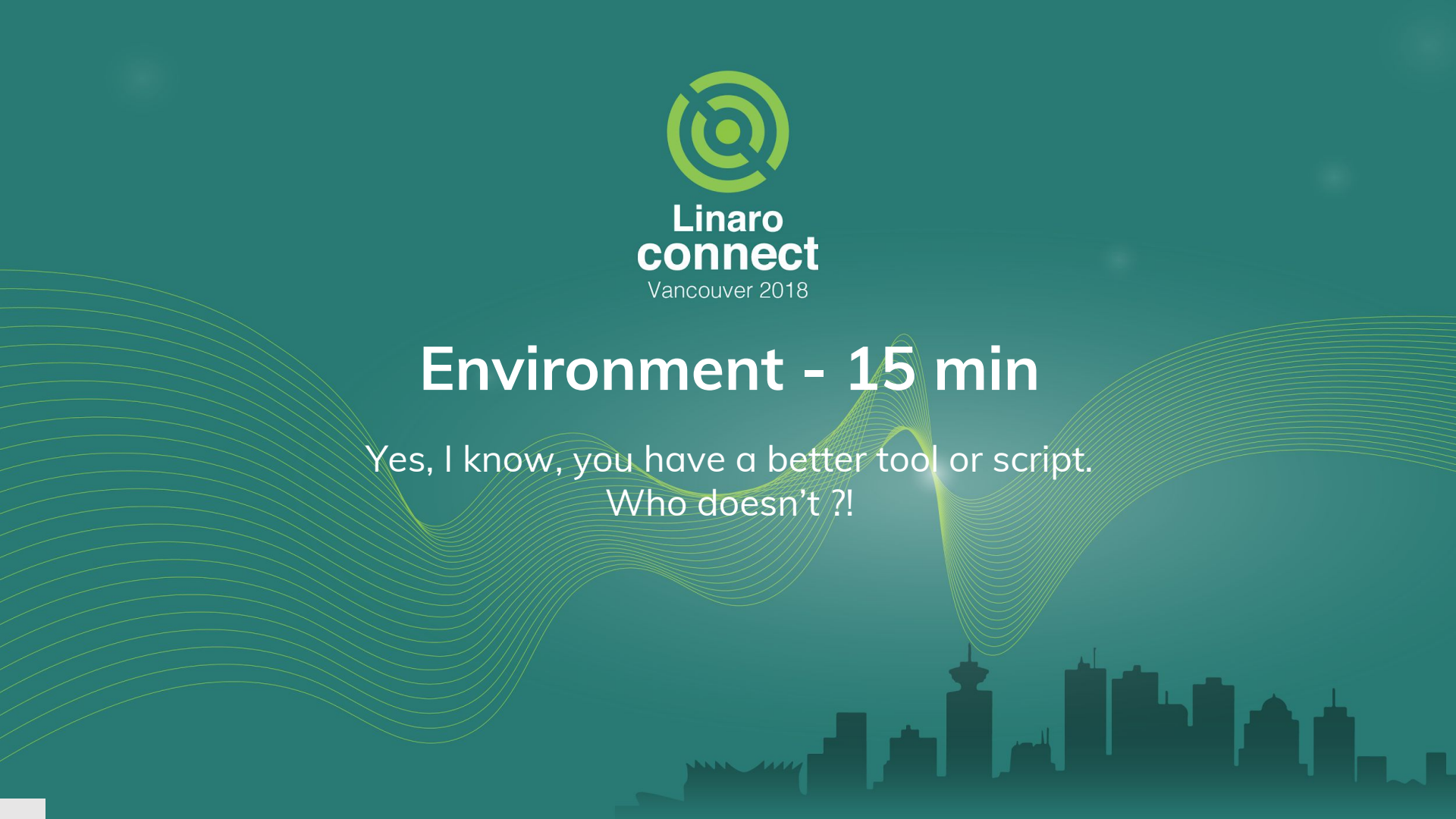
Presentation Video: https://lnkd.in/dxUyjbV

Topics to be covered

- **Environment**
  - Work directory “idea”
  - X-compiling & containers
  - Package generation and repository
  - Bug reproduction environment
    - KVM/QEMU guests
    - Boards
    - Packaged Test Suites
- **Eclipse CDT as IDE**
  - Using Eclipse as IDE
  - Debugging with Eclipse
  - **Eclipse** as IDE for the Linux Kernel
- **Debugging: Real cases**
  - Interpreting Issues
    - Foundations
    - Dead locks / Race conditions
    - Usual issues w/ Kernel bugs
    - Stack traces
  - **BUG #3303**
    - Using crash

Who am I ?





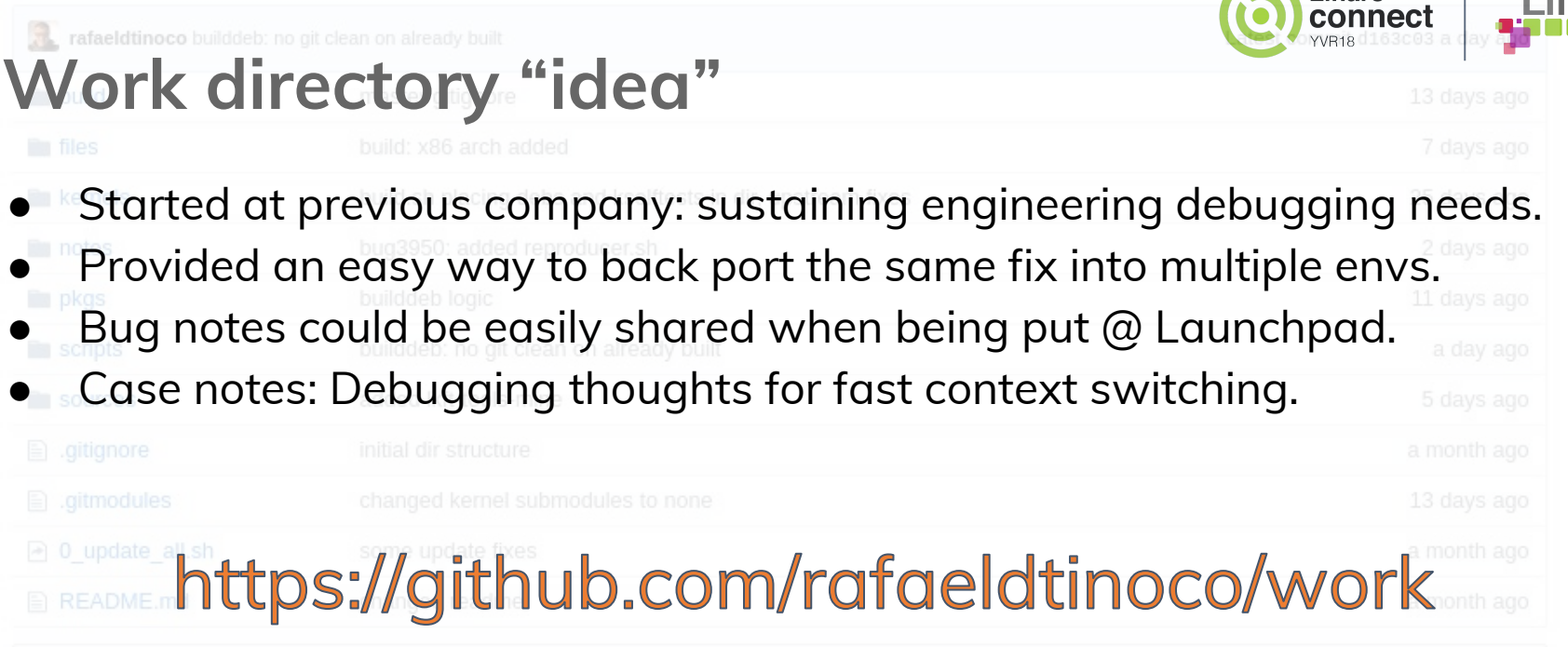
Environment - 15 min

Yes, I know, you have a better tool or script.
Who doesn't ?!

Work directory “idea”

- Started at previous company: sustaining engineering debugging needs.
- Provided an easy way to back port the same fix into multiple envs.
- Bug notes could be easily shared when being put @ Launchpad.
- Case notes: Debugging thoughts for fast context switching.

https://github.com/rafaeldtinoco/work



X-compiling and containers

- Containers
  - **QEMU-user-static** for arm
    - sysctl -a fs.binfmt_misc | grep qemu
  - LXC backed by **debootstrap**
  - lxc-create
    -t download --name mytemplate --
    -d debian -r sid
    -a (i386|amd64|armhf|arm64)
  - isc-dhcp-server AND bind9 (dynamic dns)

```
inaddy@workstation:~$ time lxc-copy -B overlayfs -s -n worklxcamd64 -N lxc01

real    0m0.104s
user    0m0.009s
sys     0m0.006s
inaddy@workstation:~$ lxc-start -n lxc01
inaddy@workstation:~$ ssh lxc01
(c)inaddy@lxc01:~$
```

  - **shared mounts** between host and LXC

```
inaddy@workstation:~$ ls ~/work
0_update_all.sh  build  files  kernels  notes  pkgs  README.md  scratch  scripts  sources
inaddy@workstation:~$ ssh lxc01
(c)inaddy@lxc01:~$ ls ~/work
0_update_all.sh  build  files  kernels  notes  pkgs  README.md  scratch  scripts  sources
(c)inaddy@lxc01:~$ s
```

- X-compiling w/ QEMU-user-static
  - When gcc/libc cross comp. isn't enough
  - Package dependencies are fully met
  - **chroots + qemu-user-static is enough**
  - chroots:
    - **aren't good for services**
    - aren't good for "dist-upgrades"
      - if mounting VM ext4 loop dev
    - won't give you diff namespaces
  - LXC has some issues, but... it is **usable**
    - armhf or arm64 container on x86
  - QEMU static is **faster** than QEMU VM

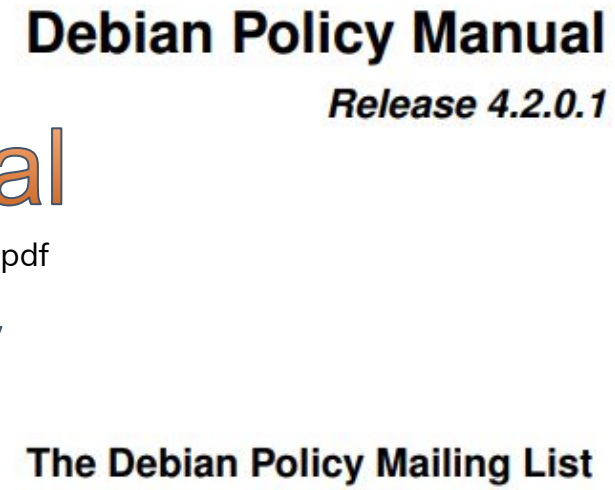
Package generation and repository

Debian Packaging Tutorial

https://www.debian.org/doc/manuals/packaging-tutorial/packaging-tutorial.en.pdf

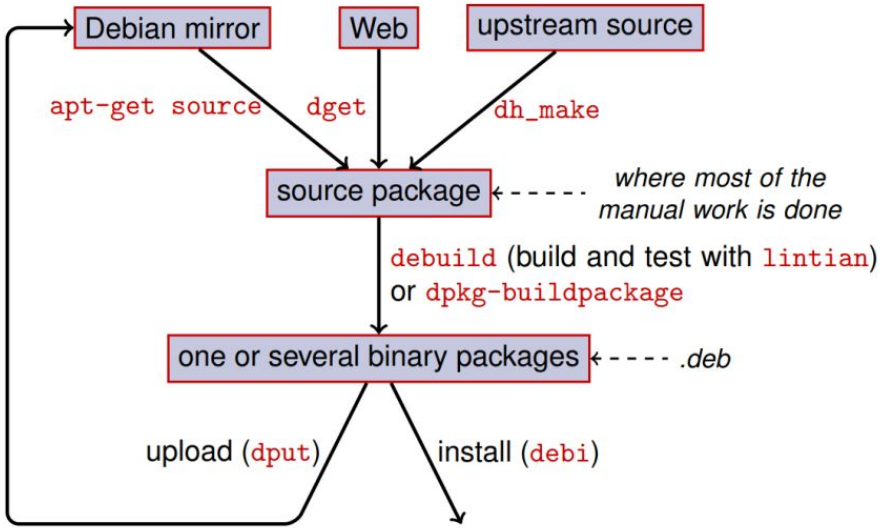
Debian Packaging Policy

https://www.debian.org/doc/debian-policy/policy.pdf



Package generation and repository

- Debian package workflow:



- Basic idea:

```
$ apt-get install build-essential
$ apt-get install devscripts
$ apt-get install ubuntu-dev-tools

$ apt-get build-dep <package>
```

- And then...

```
$ cd <package_dir>
$ fakeroot debian/rules binary
```

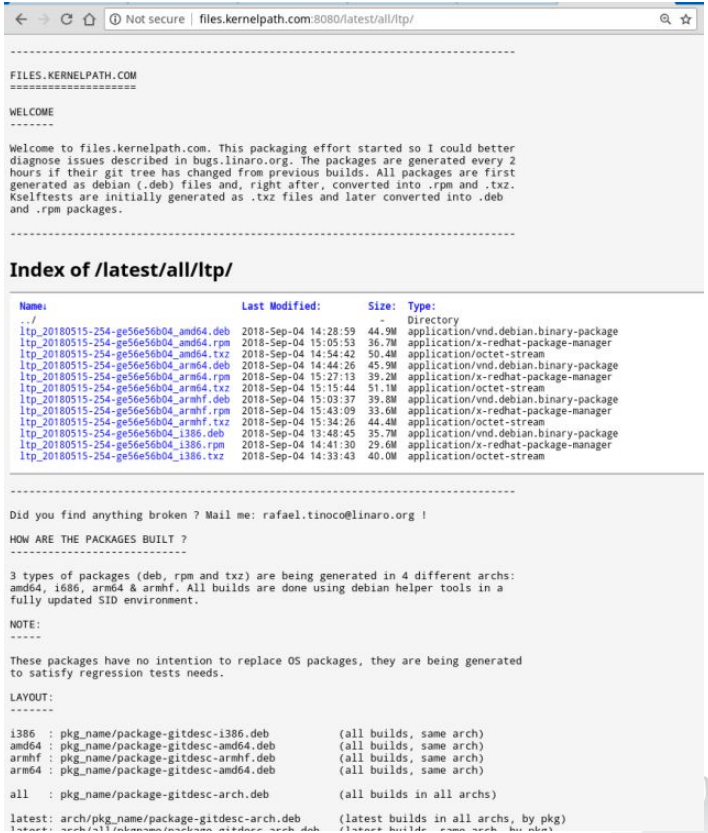
or

```
$ debuild -b -us -uc
```

DEMO

Credits for Slide Content: Lucas Nussbaum (packaging-tutorial@packages.debian.org)

BUG reproduction environment: Boards: Pkg generation and repo



Not secure | files.kernelpath.com:8080/latest/all/ltp/

FILES.KERNELPATH.COM
==========

WELCOME

Welcome to files.kernelpath.com. This packaging effort started so I could better diagnose issues described in bugs.linaro.org. The packages are generated every 2 hours if their git tree has changed from previous builds. All packages are first generated as debian (.deb) files and, right after, converted into .rpm and .txz. Kselftests are initially generated as .txz files and later converted into .deb and .rpm packages.

Index of /latest/all/ltp/

Name:	Last Modified:	Size:	Type:
../		-	Directory
ltp_20180515-254-ge56e56b04_amd64.deb	2018-Sep-04 14:28:59	44.9M	application/vnd.debian.binary-package
ltp_20180515-254-ge56e56b04_amd64.rpm	2018-Sep-04 15:05:53	36.7M	application/x-redhat-package-manager
ltp_20180515-254-ge56e56b04_amd64.txz	2018-Sep-04 14:54:42	50.4M	application/octet-stream
ltp_20180515-254-ge56e56b04_arm64.deb	2018-Sep-04 14:44:26	45.9M	application/vnd.debian.binary-package
ltp_20180515-254-ge56e56b04_arm64.rpm	2018-Sep-04 15:27:13	39.2M	application/x-redhat-package-manager
ltp_20180515-254-ge56e56b04_arm64.txz	2018-Sep-04 15:15:44	51.1M	application/octet-stream
ltp_20180515-254-ge56e56b04_armhf.deb	2018-Sep-04 15:03:37	39.8M	application/vnd.debian.binary-package
ltp_20180515-254-ge56e56b04_armhf.rpm	2018-Sep-04 15:43:09	33.6M	application/x-redhat-package-manager
ltp_20180515-254-ge56e56b04_armhf.txz	2018-Sep-04 15:34:26	44.4M	application/octet-stream
ltp_20180515-254-ge56e56b04_i386.deb	2018-Sep-04 13:48:45	35.7M	application/vnd.debian.binary-package
ltp_20180515-254-ge56e56b04_i386.rpm	2018-Sep-04 14:41:30	29.6M	application/x-redhat-package-manager
ltp_20180515-254-ge56e56b04_i386.txz	2018-Sep-04 14:33:43	40.0M	application/octet-stream

---

Did you find anything broken ? Mail me: rafael.tinoco@linaro.org !

HOW ARE THE PACKAGES BUILT ?
----

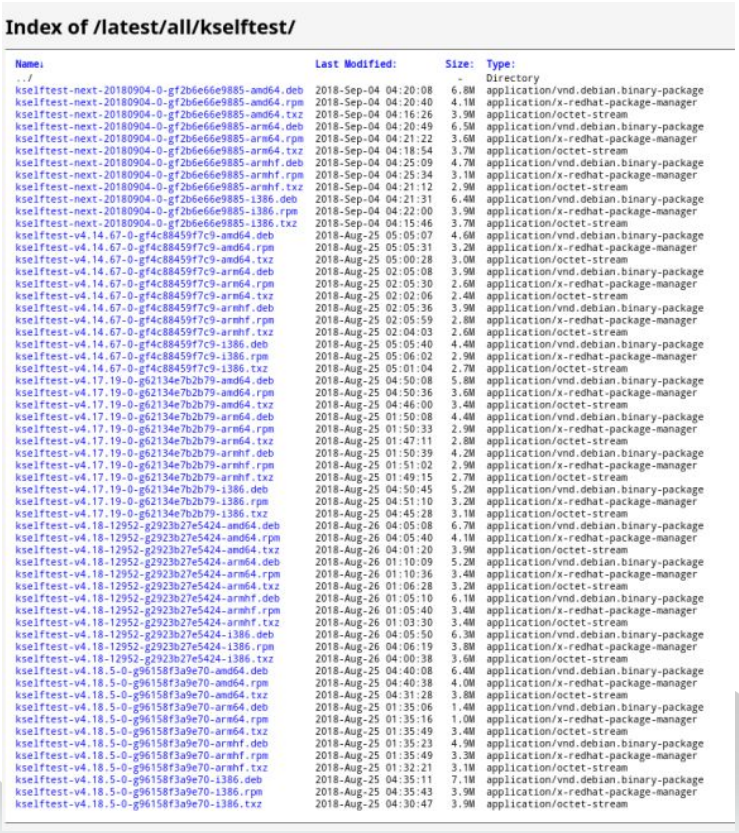
3 types of packages (deb, rpm and txz) are being generated in 4 different archs: amd64, i686, arm64 & armhf. All builds are done using debian helper tools in a fully updated SID environment.

NOTE:

These packages have no intention to replace OS packages, they are being generated to satisfy regression tests needs.

LAYOUT:

```
i386  : pkg_name/package-gitdesc-i386.deb          (all builds, same arch)
amd64 : pkg_name/package-gitdesc-amd64.deb         (all builds, same arch)
armhf : pkg_name/package-gitdesc-armhf.deb         (all builds, same arch)
arm64 : pkg_name/package-gitdesc-amd64.deb         (all builds, same arch)

all   : pkg_name/package-gitdesc-arch.deb          (all builds in all archs)

latest: arch/pkg_name/package-gitdesc-arch.deb     (latest builds in all archs, by pkg)
```



Index of /latest/all/kselftest/

Name:	Last Modified:	Size:	Type:
../		-	Directory
kselftest-next-20180904-0-gf2b6e66e9885-amd64.deb	2018-Sep-04 04:20:08	6.8M	application/vnd.debian.binary-package
kselftest-next-20180904-0-gf2b6e66e9885-amd64.rpm	2018-Sep-04 04:20:40	4.1M	application/x-redhat-package-manager
kselftest-next-20180904-0-gf2b6e66e9885-amd64.txz	2018-Sep-04 04:16:26	3.9M	application/octet-stream
kselftest-next-20180904-0-gf2b6e66e9885-arm64.deb	2018-Sep-04 04:20:49	6.5M	application/vnd.debian.binary-package
kselftest-next-20180904-0-gf2b6e66e9885-arm64.rpm	2018-Sep-04 04:21:22	3.6M	application/x-redhat-package-manager
kselftest-next-20180904-0-gf2b6e66e9885-arm64.txz	2018-Sep-04 04:18:54	3.7M	application/octet-stream
kselftest-next-20180904-0-gf2b6e66e9885-armhf.deb	2018-Sep-04 04:25:09	4.7M	application/vnd.debian.binary-package
kselftest-next-20180904-0-gf2b6e66e9885-armhf.rpm	2018-Sep-04 04:25:34	3.1M	application/x-redhat-package-manager
kselftest-next-20180904-0-gf2b6e66e9885-armhf.txz	2018-Sep-04 04:21:12	2.9M	application/octet-stream
kselftest-next-20180904-0-gf2b6e66e9885-i386.deb	2018-Sep-04 04:21:31	6.4M	application/vnd.debian.binary-package
kselftest-next-20180904-0-gf2b6e66e9885-i386.rpm	2018-Sep-04 04:22:00	3.9M	application/x-redhat-package-manager
kselftest-next-20180904-0-gf2b6e66e9885-i386.txz	2018-Sep-04 04:15:46	3.7M	application/octet-stream
kselftest-v4.14.67-0-gf4c88459f7c9-amd64.deb	2018-Aug-25 05:05:07	4.6M	application/vnd.debian.binary-package
kselftest-v4.14.67-0-gf4c88459f7c9-amd64.rpm	2018-Aug-25 05:05:31	3.2M	application/x-redhat-package-manager
kselftest-v4.14.67-0-gf4c88459f7c9-amd64.txz	2018-Aug-25 05:00:28	3.0M	application/octet-stream
kselftest-v4.14.67-0-gf4c88459f7c9-arm64.deb	2018-Aug-25 02:05:08	3.9M	application/vnd.debian.binary-package
kselftest-v4.14.67-0-gf4c88459f7c9-arm64.rpm	2018-Aug-25 02:05:30	2.6M	application/x-redhat-package-manager
kselftest-v4.14.67-0-gf4c88459f7c9-arm64.txz	2018-Aug-25 02:02:06	2.4M	application/octet-stream
kselftest-v4.14.67-0-gf4c88459f7c9-armhf.deb	2018-Aug-25 02:05:36	3.9M	application/vnd.debian.binary-package
kselftest-v4.14.67-0-gf4c88459f7c9-armhf.rpm	2018-Aug-25 02:05:59	2.8M	application/x-redhat-package-manager
kselftest-v4.14.67-0-gf4c88459f7c9-armhf.txz	2018-Aug-25 02:04:03	2.6M	application/octet-stream
kselftest-v4.14.67-0-gf4c88459f7c9-i386.deb	2018-Aug-25 05:05:40	4.4M	application/vnd.debian.binary-package
kselftest-v4.14.67-0-gf4c88459f7c9-i386.rpm	2018-Aug-25 05:06:02	2.9M	application/x-redhat-package-manager
kselftest-v4.14.67-0-gf4c88459f7c9-i386.txz	2018-Aug-25 05:01:04	2.7M	application/octet-stream
kselftest-v4.17.19-0-g62134e7b2b79-amd64.deb	2018-Aug-25 04:50:08	5.8M	application/vnd.debian.binary-package
kselftest-v4.17.19-0-g62134e7b2b79-amd64.rpm	2018-Aug-25 04:50:36	3.6M	application/x-redhat-package-manager
kselftest-v4.17.19-0-g62134e7b2b79-amd64.txz	2018-Aug-25 04:46:00	3.4M	application/octet-stream
kselftest-v4.17.19-0-g62134e7b2b79-arm64.deb	2018-Aug-25 01:50:08	4.4M	application/vnd.debian.binary-package
kselftest-v4.17.19-0-g62134e7b2b79-arm64.rpm	2018-Aug-25 01:50:33	2.9M	application/x-redhat-package-manager
kselftest-v4.17.19-0-g62134e7b2b79-arm64.txz	2018-Aug-25 01:47:11	2.8M	application/octet-stream
kselftest-v4.17.19-0-g62134e7b2b79-armhf.deb	2018-Aug-25 01:50:39	4.2M	application/vnd.debian.binary-package
kselftest-v4.17.19-0-g62134e7b2b79-armhf.rpm	2018-Aug-25 01:51:02	2.9M	application/x-redhat-package-manager
kselftest-v4.17.19-0-g62134e7b2b79-armhf.txz	2018-Aug-25 01:49:15	2.7M	application/octet-stream
kselftest-v4.17.19-0-g62134e7b2b79-i386.deb	2018-Aug-25 04:50:45	5.2M	application/vnd.debian.binary-package
kselftest-v4.17.19-0-g62134e7b2b79-i386.rpm	2018-Aug-25 04:51:10	3.2M	application/x-redhat-package-manager
kselftest-v4.17.19-0-g62134e7b2b79-i386.txz	2018-Aug-25 04:45:28	3.1M	application/octet-stream
kselftest-v4.18-12952-g2923b27e5424-amd64.deb	2018-Aug-26 04:05:08	6.7M	application/vnd.debian.binary-package
kselftest-v4.18-12952-g2923b27e5424-amd64.rpm	2018-Aug-26 04:05:40	4.1M	application/x-redhat-package-manager
kselftest-v4.18-12952-g2923b27e5424-amd64.txz	2018-Aug-26 04:01:20	3.9M	application/octet-stream
kselftest-v4.18-12952-g2923b27e5424-arm64.deb	2018-Aug-26 01:10:09	5.2M	application/vnd.debian.binary-package
kselftest-v4.18-12952-g2923b27e5424-arm64.rpm	2018-Aug-26 01:10:36	3.4M	application/x-redhat-package-manager
kselftest-v4.18-12952-g2923b27e5424-arm64.txz	2018-Aug-26 01:06:28	3.2M	application/octet-stream
kselftest-v4.18-12952-g2923b27e5424-armhf.deb	2018-Aug-26 01:05:10	6.1M	application/vnd.debian.binary-package
kselftest-v4.18-12952-g2923b27e5424-armhf.rpm	2018-Aug-26 01:05:40	3.4M	application/x-redhat-package-manager
kselftest-v4.18-12952-g2923b27e5424-armhf.txz	2018-Aug-26 01:03:30	3.4M	application/octet-stream
kselftest-v4.18-12952-g2923b27e5424-i386.deb	2018-Aug-26 04:05:50	6.3M	application/vnd.debian.binary-package
kselftest-v4.18-12952-g2923b27e5424-i386.rpm	2018-Aug-26 04:06:19	3.8M	application/x-redhat-package-manager
kselftest-v4.18-12952-g2923b27e5424-i386.txz	2018-Aug-26 04:00:38	3.6M	application/octet-stream
kselftest-v4.18.5-0-g96158f3a9e70-amd64.deb	2018-Aug-25 04:40:08	6.4M	application/vnd.debian.binary-package
kselftest-v4.18.5-0-g96158f3a9e70-amd64.rpm	2018-Aug-25 04:40:38	4.0M	application/x-redhat-package-manager
kselftest-v4.18.5-0-g96158f3a9e70-amd64.txz	2018-Aug-25 04:31:28	3.8M	application/octet-stream
kselftest-v4.18.5-0-g96158f3a9e70-arm64.deb	2018-Aug-25 01:35:06	1.4M	application/vnd.debian.binary-package
kselftest-v4.18.5-0-g96158f3a9e70-arm64.rpm	2018-Aug-25 01:35:16	1.0M	application/x-redhat-package-manager
kselftest-v4.18.5-0-g96158f3a9e70-arm64.txz	2018-Aug-25 01:35:49	3.4M	application/octet-stream
kselftest-v4.18.5-0-g96158f3a9e70-armhf.deb	2018-Aug-25 01:35:23	4.9M	application/vnd.debian.binary-package
kselftest-v4.18.5-0-g96158f3a9e70-armhf.rpm	2018-Aug-25 01:35:49	3.3M	application/x-redhat-package-manager
kselftest-v4.18.5-0-g96158f3a9e70-armhf.txz	2018-Aug-25 01:32:21	3.1M	application/octet-stream
kselftest-v4.18.5-0-g96158f3a9e70-i386.deb	2018-Aug-25 04:35:11	7.1M	application/vnd.debian.binary-package
kselftest-v4.18.5-0-g96158f3a9e70-i386.rpm	2018-Aug-25 04:35:43	3.9M	application/x-redhat-package-manager
kselftest-v4.18.5-0-g96158f3a9e70-i386.txz	2018-Aug-25 04:30:47	3.9M	application/octet-stream

BUG reproduction environment: KVM & QEMU guests


```
inaddy@workstation:~$ virtclone.sh workkvmamd64 bug000test01
running:
- qcowhostname.sh bug000test01
bug000test01
- qcowhome.sh bug000test01
sending home files to lxc3438 (bug000test01)...
inaddy@workstation:~$ virtclone.sh workkvmamd64 bug000test02
running:
- qcowhostname.sh bug000test02
bug000test02
- qcowhome.sh bug000test02
sending home files to lxc3855 (bug000test02)...
```

```
inaddy@workstation:~$ virsh start --console bug000test01
Domain bug000test01 started
Connected to domain bug000test01
Escape character is ^]
[    0.000000] Linux version 4.17.0-3-amd64 (debian-kernel@lists.debian.org) (gc
MP Debian 4.17.17-1 (2018-08-18)
[    0.000000] Command line: root=/dev/vda noresume console=tty0 console=ttyS0,3
rnel=256M
[    0.000000] random: get_random_u32 called from bsp_init_amd+0x25d/0x2a0 with
[    0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point regi
[    0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[    0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[    0.000000] x86/fpu: xstate_offset[2]:  576, xstate_sizes[2]:  256
[    0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes,
[    0.000000] e820: BIOS-provided physical RAM map:
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
[    0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
```

```
inaddy@workstation:~$ virsh start bug000test02
Domain bug000test02 started

inaddy@workstation:~$ ssh bug000test02
(k)inaddy@bug000test02:~$ uname -a
Linux bug000test02 4.17.0-3-amd64 #1 SMP Debian 4.17.17-1
(k)inaddy@bug000test02:~$
```

```
inaddy@workstation:~/work/kernels/amd64/stable$ find .
.
./stable-linux-4.9.y
./stable-linux-4.9.y/linux-firmware-image-4.9.120_4.9.120-1_amd64.deb
./stable-linux-4.9.y/linux-headers-4.9.120_4.9.120-1_amd64.deb
./stable-linux-4.9.y/linux-libc-dev_4.9.120-1_amd64.deb
./stable-linux-4.9.y/linux-image-4.9.120_4.9.120-1_amd64.deb
./stable-linux-4.14.y
./stable-linux-4.14.y/linux-image-4.14.67_4.14.67-1_amd64.deb
./stable-linux-4.14.y/linux-libc-dev_4.14.67-1_amd64.deb
./stable-linux-4.14.y/linux-headers-4.14.67_4.14.67-1_amd64.deb
./stable-linux-4.4.y
./stable-linux-4.4.y/linux-image-4.4.148_4.4.148-1_amd64.deb
./stable-linux-4.4.y/linux-firmware-image-4.4.148_4.4.148-1_amd64.deb
./stable-linux-4.4.y/linux-headers-4.4.148_4.4.148-1_amd64.deb
./stable-linux-4.4.y/linux-libc-dev_4.4.148-1_amd64.deb
./stable-linux-4.18.y
./stable-linux-4.16.y
./stable-linux-4.16.y/linux-image-4.16.18_4.16.18-1_amd64.deb
./stable-linux-4.16.y/linux-headers-4.16.18_4.16.18-1_amd64.deb
./stable-linux-4.16.y/linux-libc-dev_4.16.18-1_amd64.deb
./stable-linux-4.17.y
./stable-linux-4.17.y/linux-image-4.17.19_4.17.19-1_amd64.deb
./stable-linux-4.17.y/linux-headers-4.17.19_4.17.19-1_amd64.deb
./stable-linux-4.17.y/linux-libc-dev_4.17.19-1_amd64.deb
inaddy@workstation:~/work/kernels/amd64/stable$ cd stable-linux-4.4.y
inaddy@workstation:~/work/kernels/amd64/stable/stable-linux-4.4.y$ qcowkerninst.sh bug000test02
Selecting previously unselected package linux-firmware-image-4.4.148.
(Reading database ... 123718 files and directories currently installed.)
Preparing to unpack .../linux-firmware-image-4.4.148_4.4.148-1_amd64.deb ...
Unpacking linux-firmware-image-4.4.148 (4.4.148-1) ...
Setting up linux-firmware-image-4.4.148 (4.4.148-1) ...
Selecting previously unselected package linux-headers-4.4.148.
(Reading database ... 123869 files and directories currently installed.)
Preparing to unpack .../linux-headers-4.4.148_4.4.148-1_amd64.deb ...
Unpacking linux-headers-4.4.148 (4.4.148-1) ...
Setting up linux-headers-4.4.148 (4.4.148-1) ...
Selecting previously unselected package linux-image-4.4.148.
(Reading database ... 141401 files and directories currently installed.)
Preparing to unpack .../linux-image-4.4.148_4.4.148-1_amd64.deb ...
Unpacking linux-image-4.4.148 (4.4.148-1) ...
Setting up linux-image-4.4.148 (4.4.148-1) ...
update-initramfs: Generating /boot/initrd.img-4.4.148
```

BUG reproduction environment: KVM & QEMU guests

```
inaddy@workstation:~$ qcowshell.sh list
bug000test01
bug000test02
bug3771kern414
bug3771kern416
bug3771kern417
bug3771kern44
bug3771kern49
bug3771kernmain
bug3771kernnext
workkvmamd64
workkvmamd64stretch
workkvmi686
workqemuamd64
workqemuarm64
workqemuarmhf
workqemui686
inaddy@workstation:~$ qcowshell.sh bug000test02
systemd 239 running in system mode. (+PAM +AUDIT +SELINUX +IMA +APPARMOR +SMA
+GNUTLS +ACL +XZ +LZ4 +SECCOMP +BLKID +ELFUTILS +KMOD -IDN2 +IDN -PCRE2 defau
Detected virtualization lxc.
Detected architecture x86.

Welcome to Debian GNU/Linux buster/sid!

Set hostname to <bug000test02>.
[  OK  ] Reached target Swap.
system.slice: Failed to reset devices.list: Operation not permitted
system-getty.slice: Failed to reset devices.list: Operation not permitted
[  OK  ] Created slice system-getty.slice.
[  OK  ] Started Dispatch Password Requests to Console Directory Watch.
[  OK  ] Listening on Journal Socket.
systemd-remount-fs.service: Failed to reset devices.list: Operation not permi
         Starting Remount Root and Kernel File Systems...
[  OK  ] Reached target Remote File Systems.
[  OK  ] Listening on Journal Socket (/dev/log).
[  OK  ] Created slice User and Session Slice.
```

```
         Starting Update UTMP about System Runlevel Changes...
[  OK  ] Started Update UTMP about System Runlevel Changes.

Debian GNU/Linux buster/sid bug000test02 console

bug000test02 login: root
Password:
[root@bug000test02:~]$ uname -a
Linux bug000test02 4.16.0-2-amd64 #1 SMP Debian 4.16.16-2 (2018-
[root@bug000test02:~]$ # this is host
[root@bug000test02:~]$
```

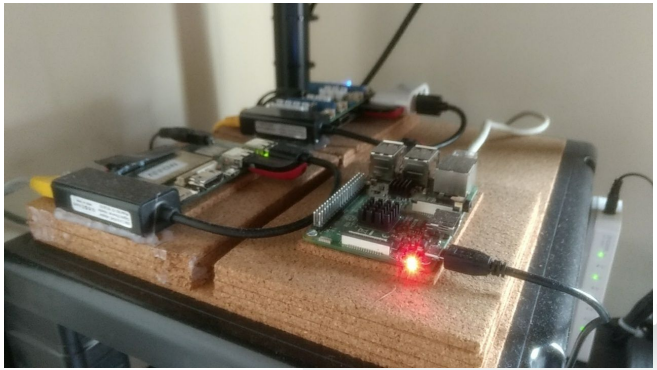
```
inaddy@workstation:~$ virsh list --all | grep bug000
 -     bug000test01                   shut off
 -     bug000test02                   shut off
inaddy@workstation:~$ virtdel.sh list
bug000test01
bug000test02
bug3771kern414
bug3771kern416
bug3771kern417
bug3771kern44
bug3771kern49
bug3771kernmain
bug3771kernnext
workkvmamd64
workkvmamd64stretch
workkvmi686
workqemuamd64
workqemuarm64
workqemuarmhf
workqemui686
inaddy@workstation:~$ virtdel.sh bug000test01
inaddy@workstation:~$ virtdel.sh bug000test02
inaddy@workstation:~$ virtdel.sh list
bug3771kern414
bug3771kern416
bug3771kern417
bug3771kern44
bug3771kern49
bug3771kernmain
bug3771kernnext
workkvmamd64
workkvmamd64stretch
workkvmi686
workqemuamd64
workqemuarm64
workqemuarmhf
workqemui686
inaddy@workstation:~$
```

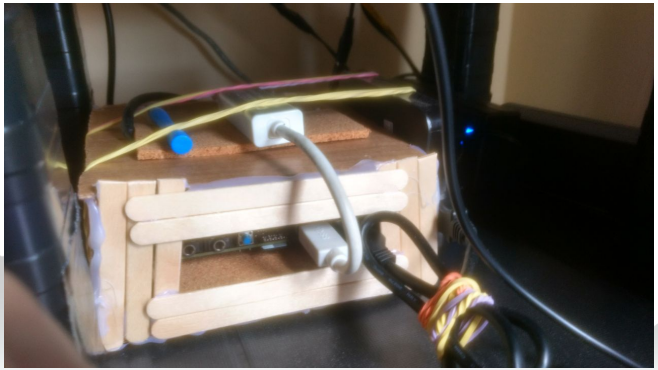
```
inaddy@workstation:~$ qcowvmlinuz.sh bug000test02 4.14
bringing lxc4680 (bug000test02) kernel/ramdisk to host
inaddy@workstation:~$ virsh start bug000test02
Domain bug000test02 started

inaddy@workstation:~$ ssh bug000test02
(k)inaddy@bug000test02:~$ uname -a
Linux bug000test02 4.14.67 #1 SMP Sun Sep 2 15:24:34 -03 201
(k)inaddy@bug000test02:~$
```

BUG reproduction environment: Boards

Cork is your friend

- Board Environment
  - workstation connected to:
    - hikey 960
    - dragonboard 410c
    - beagleboard x15
  -
```
# boards
alias hikey="ssh hikey"
alias hikeycons="sudo screen /dev/ttyUSB1 115200"
alias beagle="ssh beagle"
alias beaglecons="sudo screen /dev/ttyUSB0 115200"
alias dragon="ssh dragon"
alias dragoncons="sudo screen /dev/ttyUSB2 115200"
```



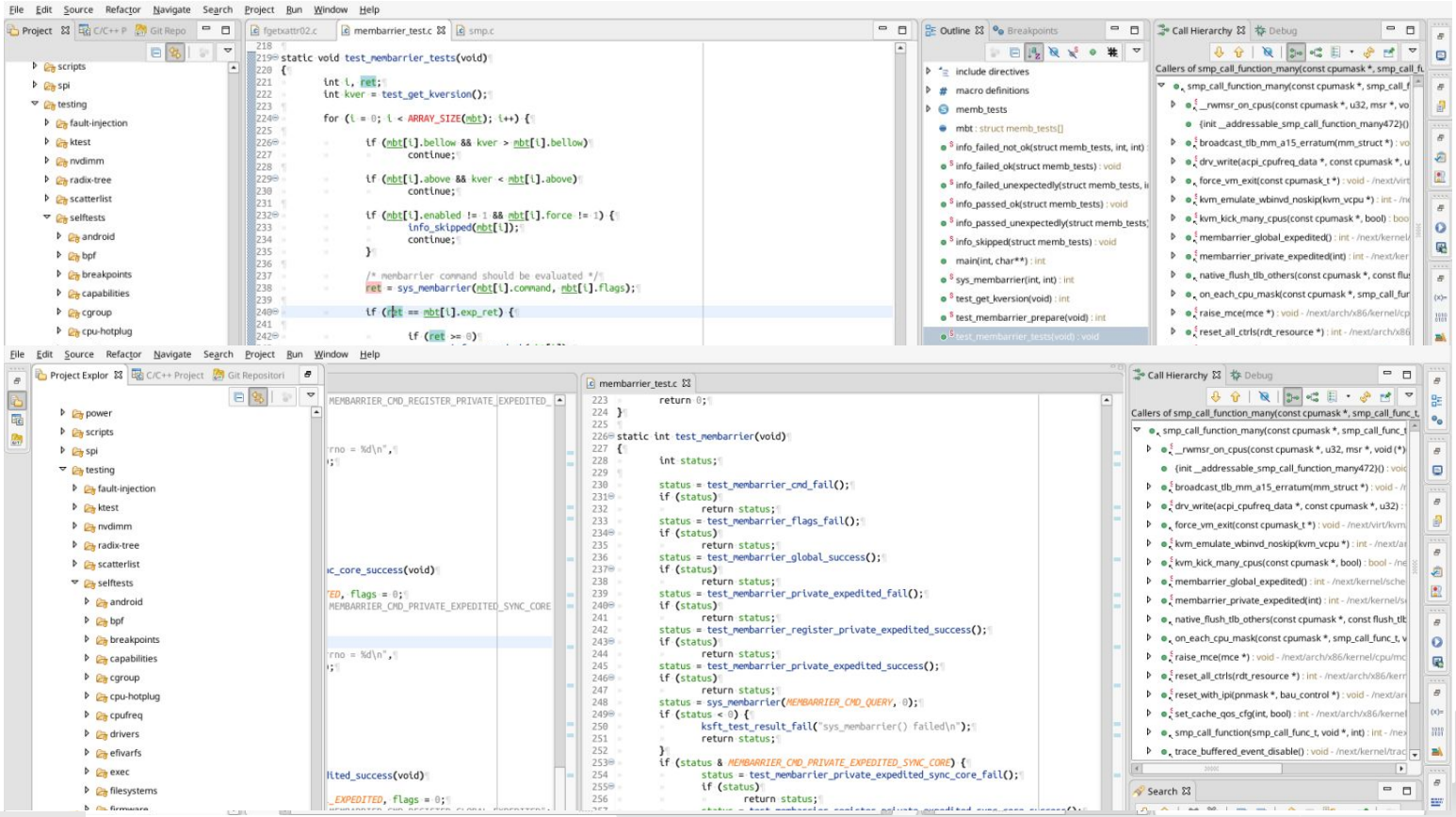


BUG reproduction environment: Boards

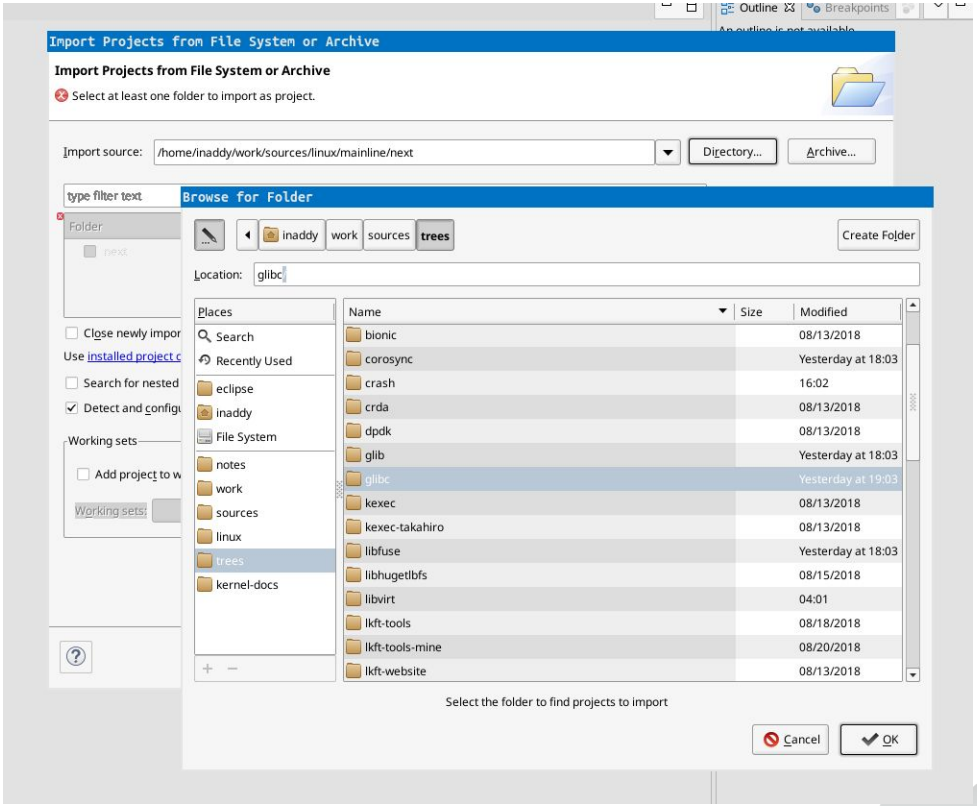
- SD cards w/ Debian installed
- Upgrading kernel w/ .deb pkgs
- Upgrading might be challenging:
  - Different recovery mechanisms
  - Different boot loaders
  - Removing SD cards manually
- QEMU/KVM, libvirt & LXC ready
- .deb test suites easily installed
- Straightforward reproduction based on LKFT output most of the times.

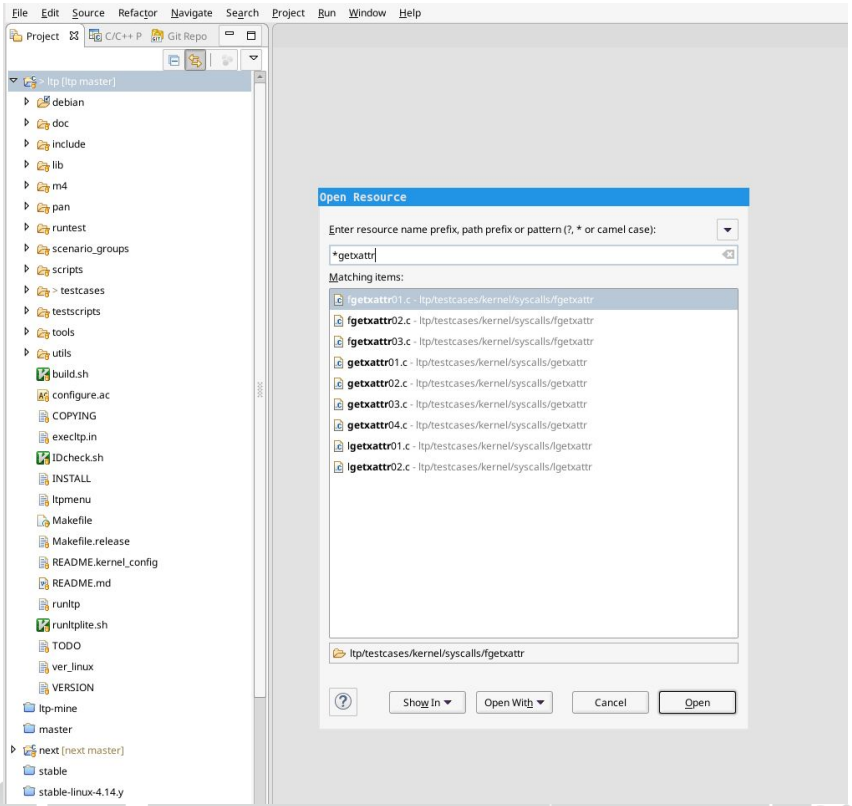

Linaro
connect
Vancouver 2018
C++
Eclipse CDT as IDE - 15 min
I know... I know...
But, for real, give it a try once or twice, you might not regret.
It happened to me...
eclipse

Using Eclipse as IDE

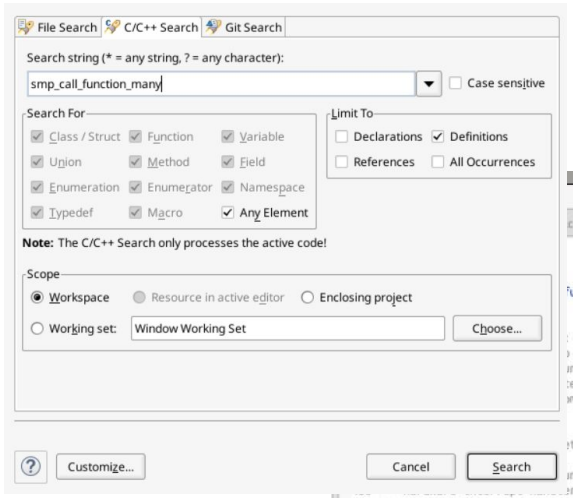


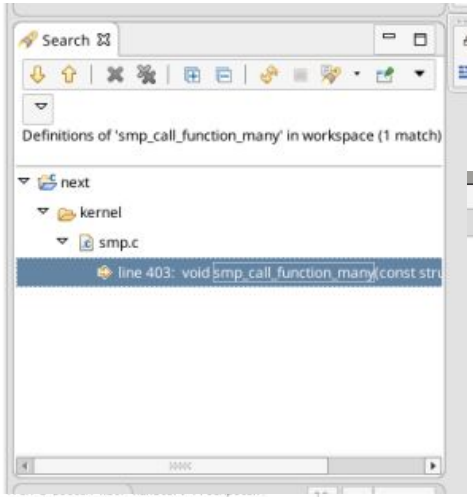
Using Eclipse as IDE



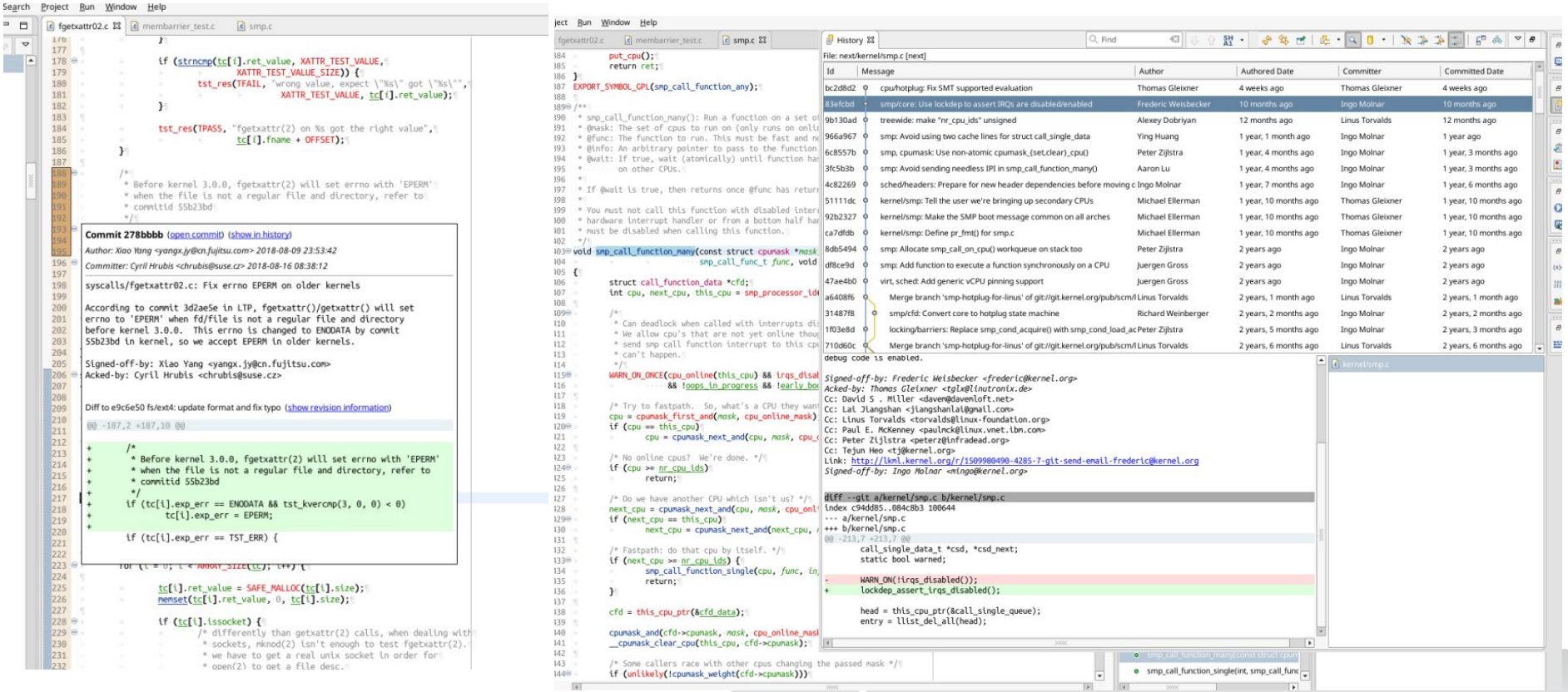


Using Eclipse as IDE

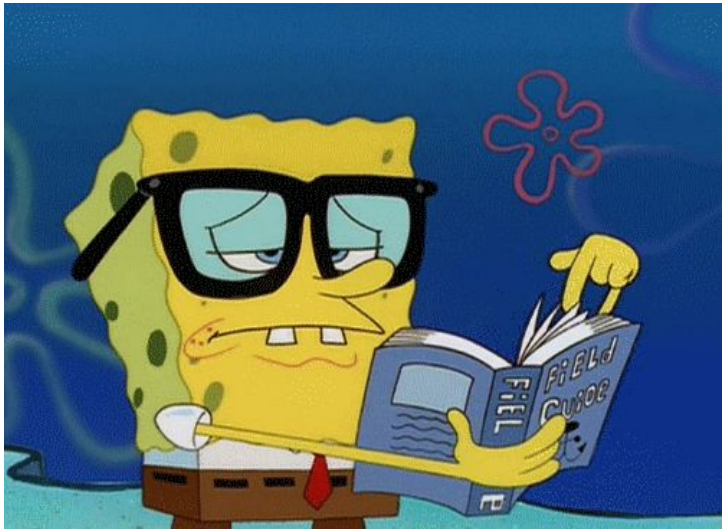




Using Eclipse as IDE



Debugging with Eclipse

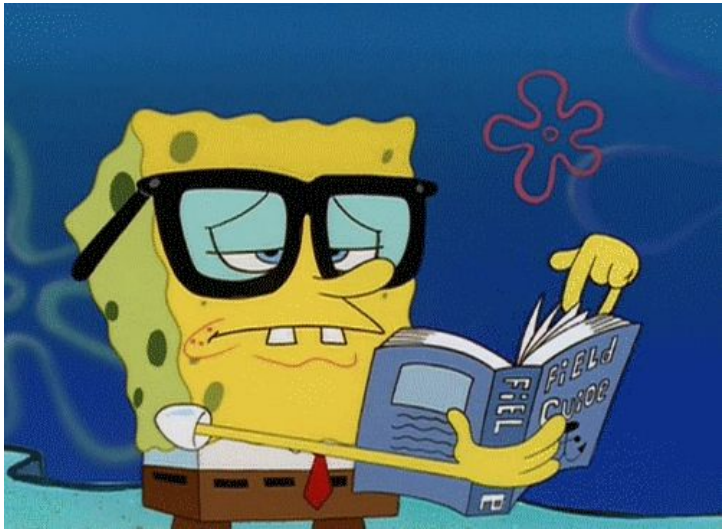


Debugging with Eclipse

- Small Example of needed effort

  Handmade Doxygen

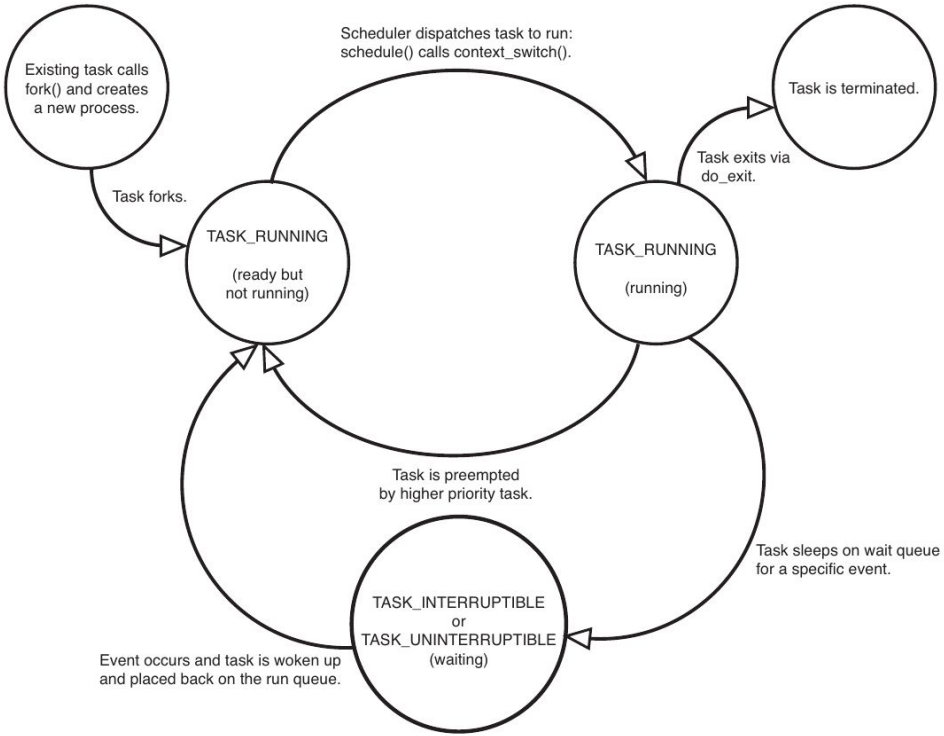
Eclipse as IDE for the Linux Kernel



Debugging: Real Cases - 60 min

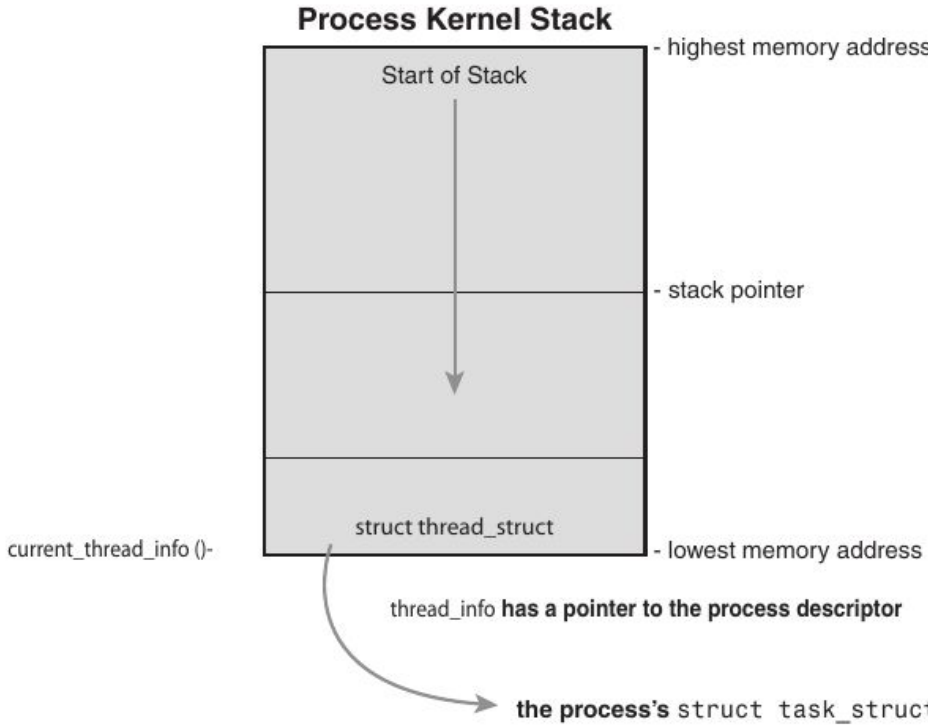
Every Bug is a learning opportunity
For Real, **Every** Bug is a learning opportunity
Trust me, **Every** Bug is a **HUGE** learning opportunity

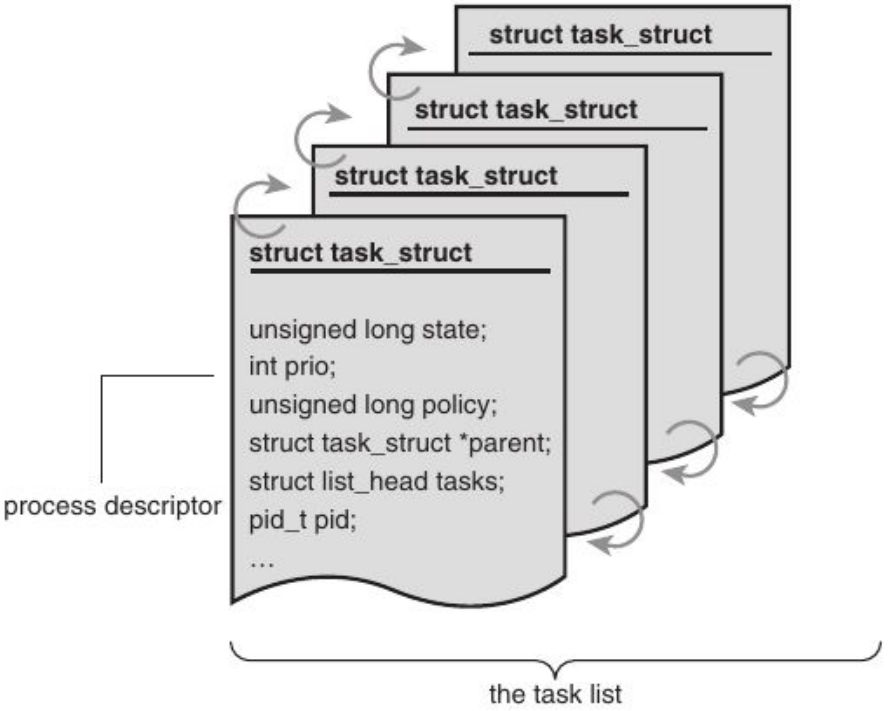
Interpreting Issues: Foundations



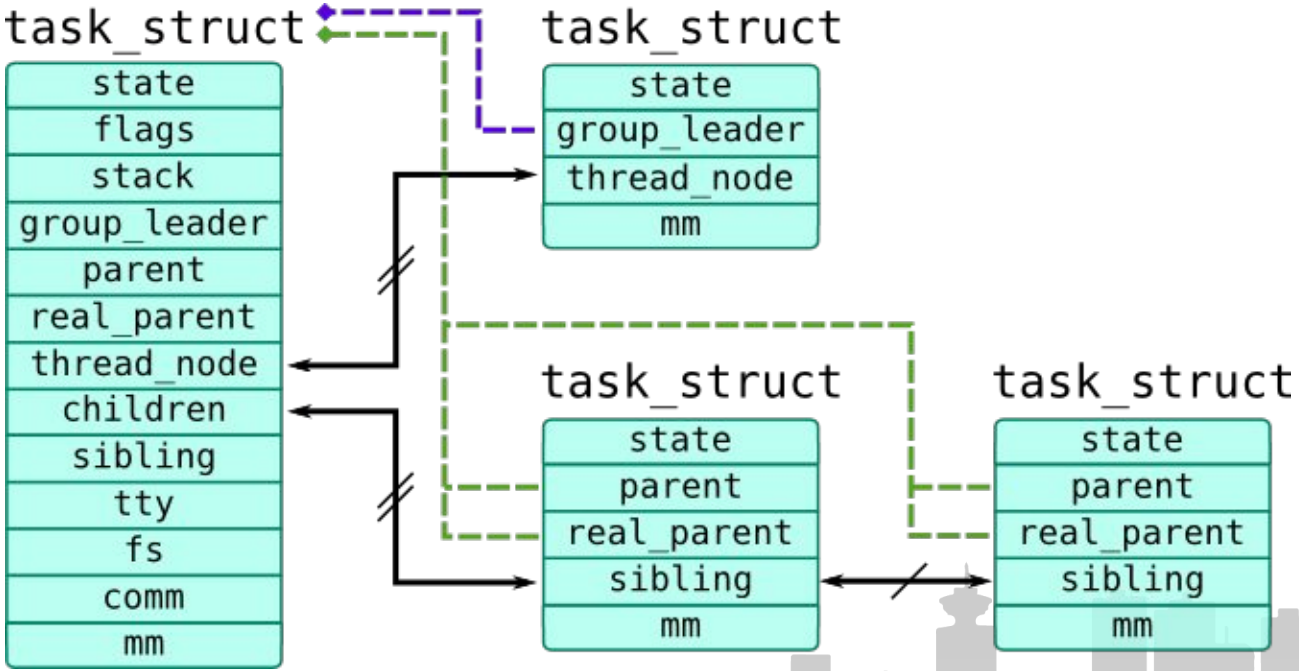
evelopment (Robert Love; Third Edition)

Interpreting Issues: Foundations





Interpreting Issues: Foundations



Interpreting Issues: Foundations

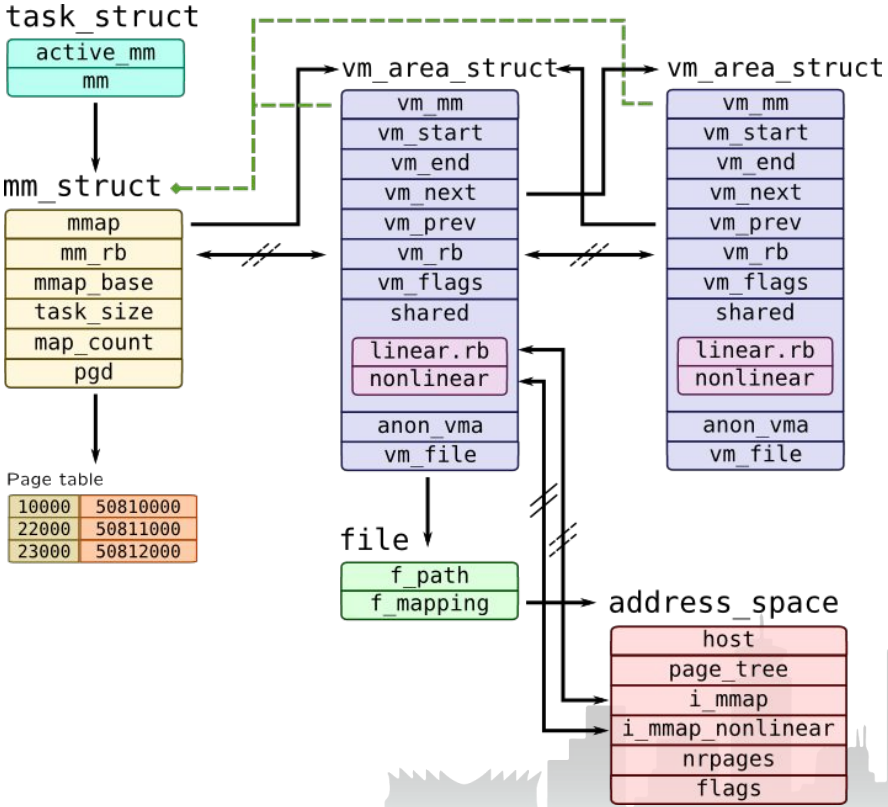
execve()

ld.so

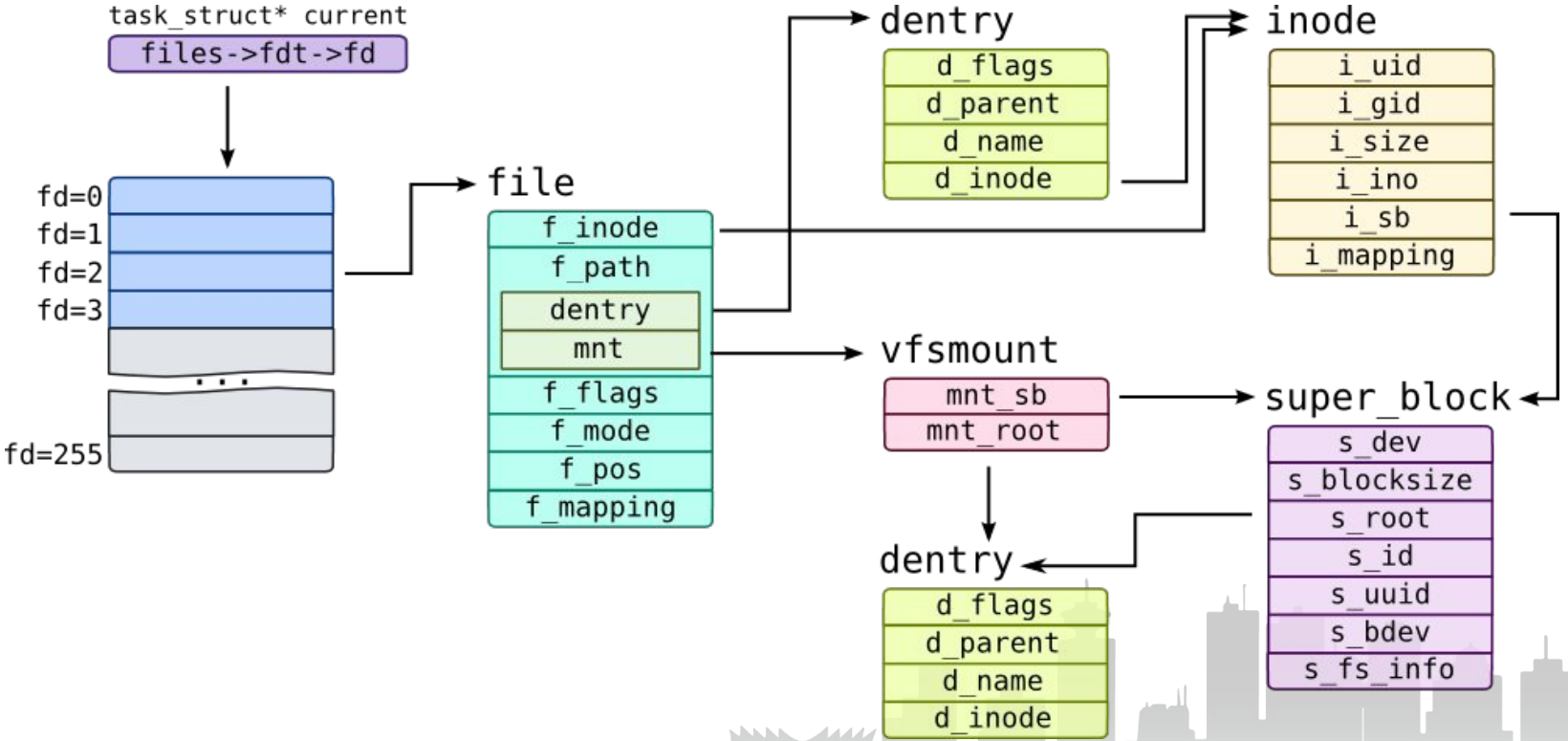
mmap()

brk()

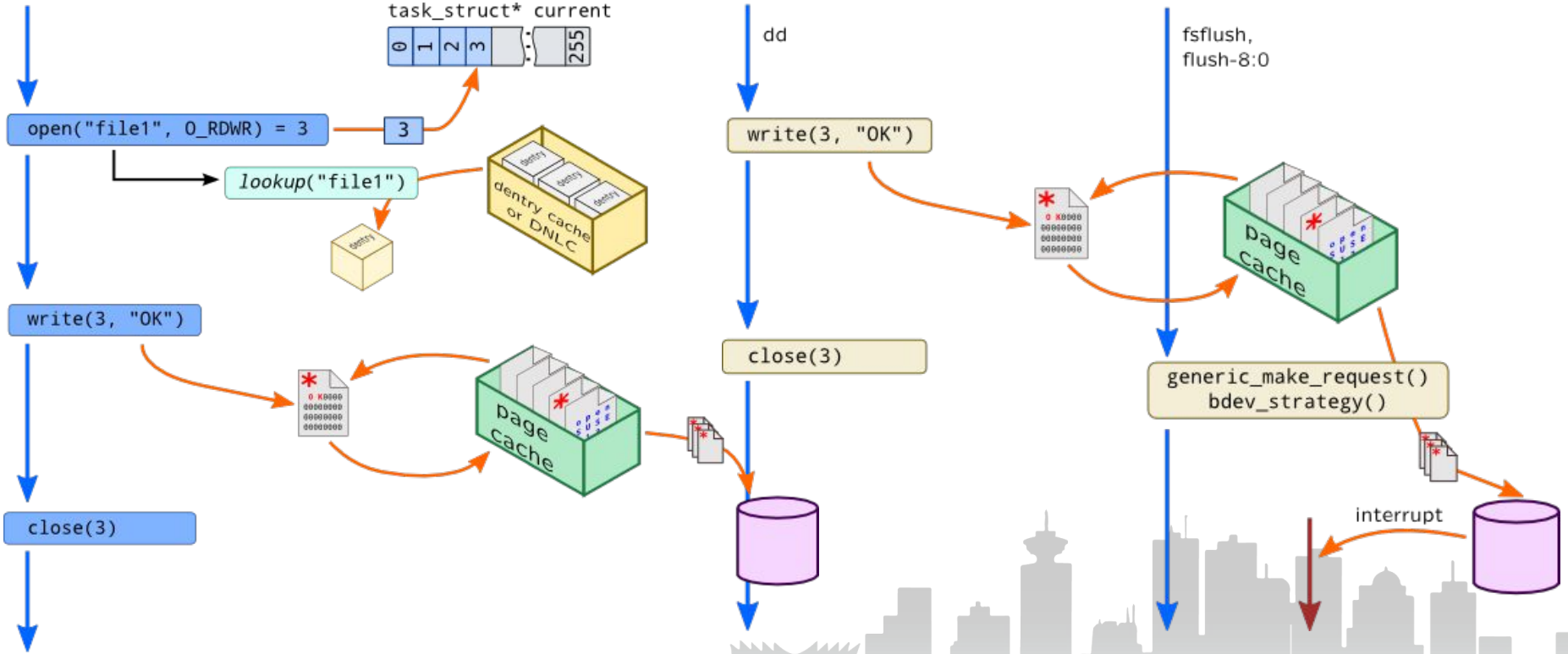
Interpreting Issues: Foundations



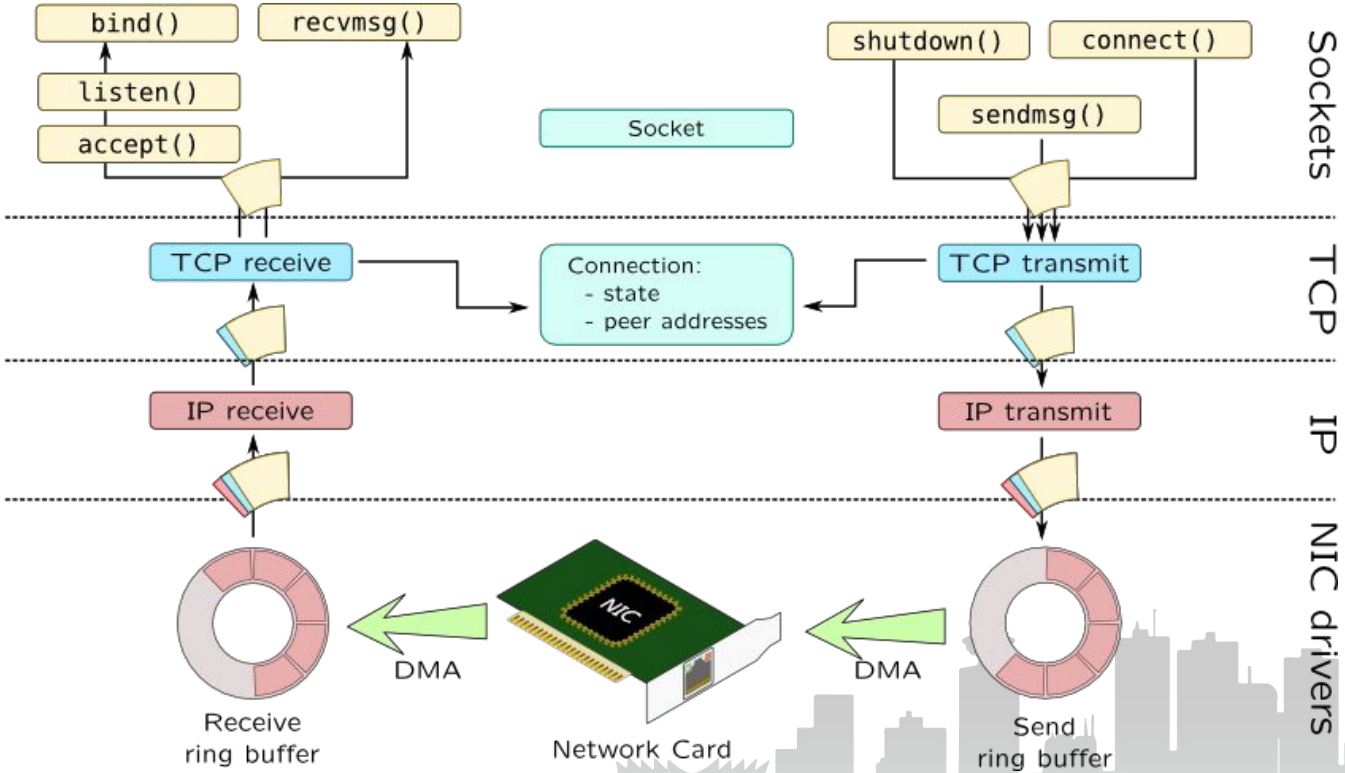
Interpreting Issues: Foundations



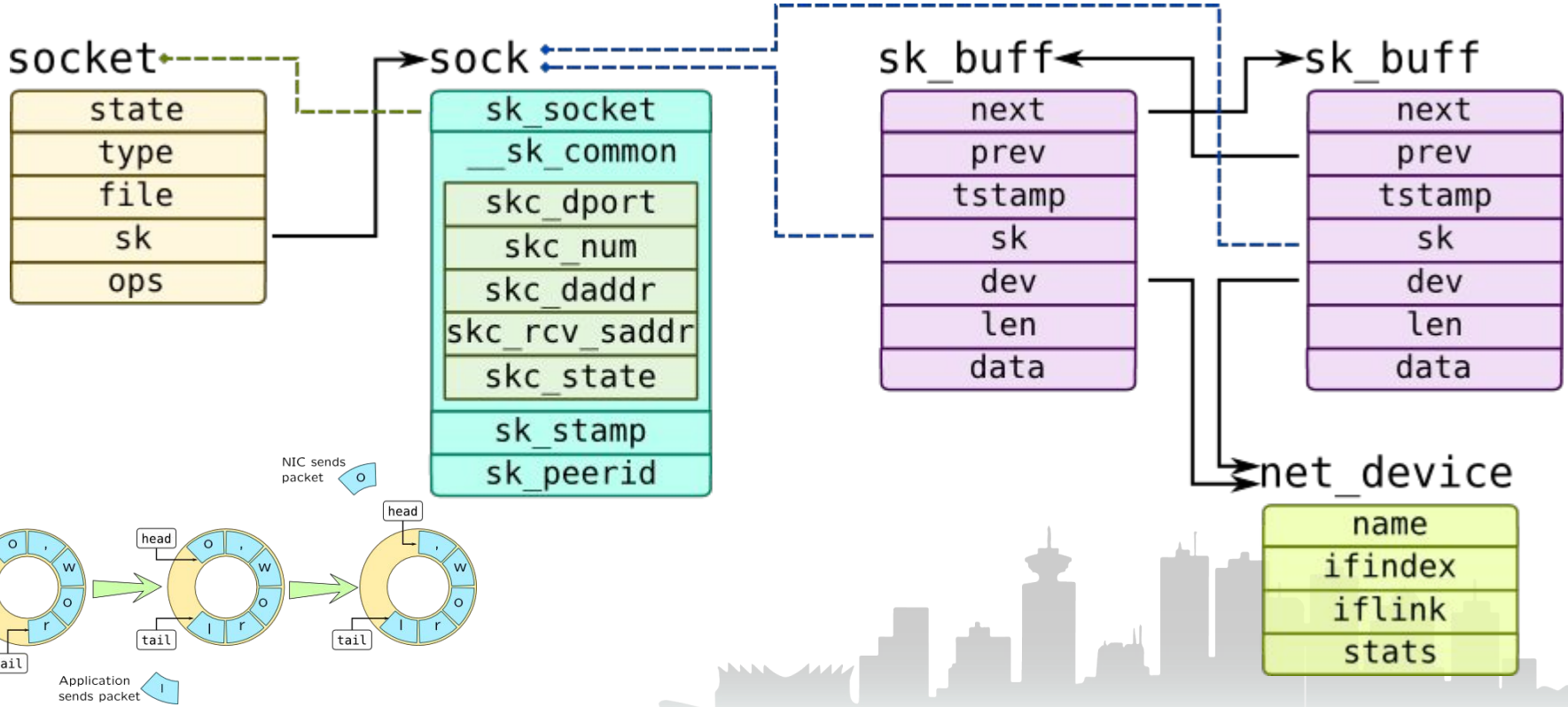
Interpreting Issues: Foundations



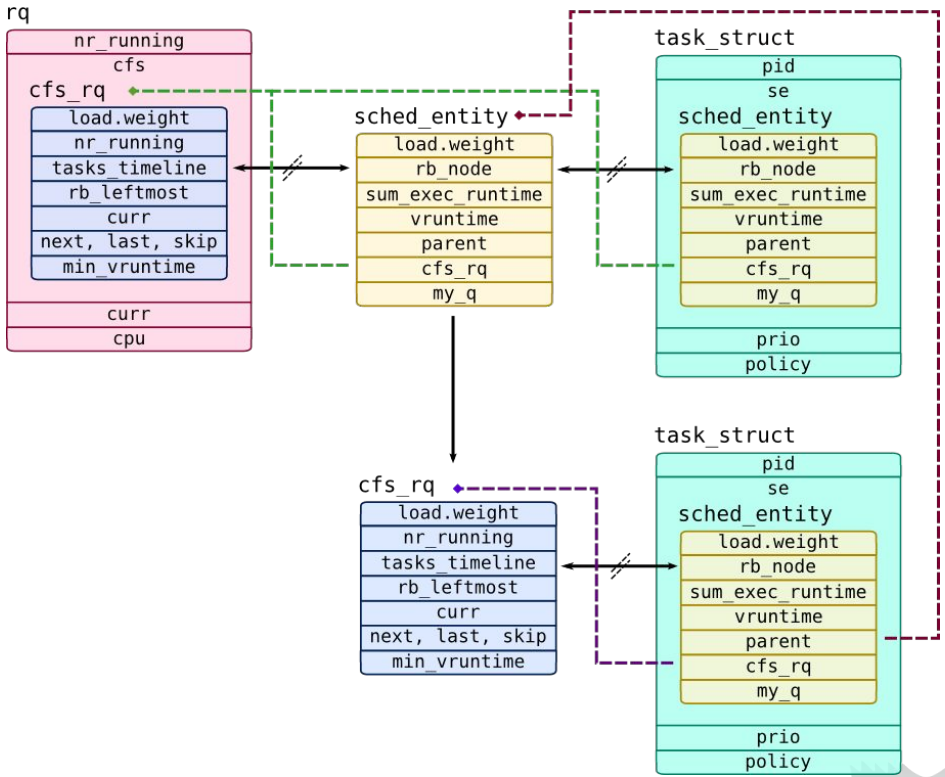
Interpreting Issues: Foundations



Interpreting Issues: Foundations



Interpreting Issues: Foundations




```
struct task_struct {
  volatile long state;
  void *stack;
  unsigned int flags;
  int prio, static_prio normal_prio;
  const struct sched_class *sched_class;
  struct sched_entity se;
  ...
};
```

```
struct sched_entity {
  struct load_weight load;
  struct rb_node run_node;
  struct list_head group_node;
  ...
};
```

```
struct ofs_rq {
  ...
  struct rb_root tasks_timeline;
  ...
};
```

```
struct rb_node {
  unsigned long rb_parent_color;
  struct rb_node *rb_right;
  struct rb_node *rb_left;
};
```

Interpreting Issues: Example of issues with Kernel messages

- Dead-locks / Inverse lock ordering warnings:
  https://bugs.linaro.org/show_bug.cgi?id=3937#c0
- Hung tasks:
  https://bugs.linaro.org/show_bug.cgi?id=3303#c6
- Different stack traces, same issue:
  https://bugs.linaro.org/show_bug.cgi?id=3903#c0 (context warn)
  https://bugs.linaro.org/show_bug.cgi?id=3903#c2 (oops)

Debugging: Real Cases
BUG #3303 (Using crash)

Overview: Starting with LTP 20170929, fanotify07 causes a kernel trace about 50% of the time.
When this happens, LTP finishes but does not finish cleanly, causing LAVA to ultimately timeout.

```
[  484.365731] INFO: task fanotify07:20858 blocked for more than 120 seconds.
[  484.372784]       Not tainted 4.9.53-rc1-00056-g3ebcc73-dirty #1
[  484.380871] \"echo 0 > /proc/sys/kernel/hung_task_timeout_secs\" disables this message.
[  484.389042] fanotify07      D    0 20858  20857 0x00000200
[  484.394741] Call trace:
[  484.397245] [<ffff000008085aec>] __switch_to+0x94/0xa8
[  484.402549] [<ffff000008a4f390>] __schedule+0x218/0xa60
[  484.407896] [<ffff000008a4fc14>] schedule+0x3c/0xa8
[  484.412931] [<ffff000008a54398>] schedule_timeout+0x1f8/0x4e8
[  484.418851] [<ffff000008a50800>] wait_for_common+0xe8/0x180
[  484.424549] [<ffff000008a508ac>] wait_for_completion+0x14/0x20
[  484.430558] [<ffff0000081384cc>] __synchronize_srcu+0x114/0x1b0
[  484.436611] [<ffff000008138590>] synchronize_srcu+0x28/0x60
[  484.442356] [<ffff0000082bde24>] fsnotify_mark_destroy_list+0x5c/0xb8
[  484.448939] [<ffff0000082bcde0>] fsnotify_destroy_group+0x38/0x68
[  484.455211] [<ffff0000082c022c>] fanotify_release+0xe4/0x118
[  484.461001] [<ffff000008271574>] __fput+0xa4/0x1e8
[  484.465946] [<ffff000008271714>] ____fput+0xc/0x18
[  484.470851] [<ffff0000080ed6dc>] task_work_run+0xcc/0x100
[  484.476415] [<ffff000008088d4c>] do_notify_resume+0xb4/0xc0
[  484.482155] [<ffff000008082ddc>] work_pending+0x8/0x14
```

Debugging: Real Cases
BUG #3303 (Kernel containing the BUG)

```
inaddy@workstation:~$ cd work/sources
inaddy@workstation:~/work/sources$ls
1_SOURCES  boards  debianizer  linux  patches  scratch  trees
inaddy@workstation:~/work/sources$cd linux
inaddy@workstation:~/work/sources/linux$ls
3_update.sh  4_build.sh  5_buildkselftest.sh  mainline  stable  stable-rc
inaddy@workstation:~/work/sources/linux$vi 4_build.sh

inaddy@workstation:~/work/sources/linux$ ./4_build.sh stable/stable-linux-4.4.y

++++++++ ENTERING stable/stable-linux-4.4.y ...
make -C /home/inaddy/work/build/linux/stable/stable-linux-4.4.y
KBUILD_SRC=/home/inaddy/work/sources/linux/stable/stable-linux-4.4.y \
-f /home/inaddy/work/sources/linux/stable/stable-linux-4.4.y/Makefile clean
make[1]: Entering directory '/home/inaddy/work/build/linux/stable/stable-linux-4.4.y'
make -f /home/inaddy/work/sources/linux/stable/stable-linux-4.4.y/scripts/Makefile.clean obj=.
make -f /home/inaddy/work/sources/linux/stable/stable-linux-4.4.y/scripts/Makefile.clean obj=arch/x86
make -f /home/inaddy/work/sources/linux/stable/stable-linux-4.4.y/scripts/Makefile.clean obj=arch/x86/lib
make -f /home/inaddy/work/sources/linux/stable/stable-linux-4.4.y/scripts/Makefile.clean obj=arch/x86/crypto
make -f /home/inaddy/work/sources/linux/stable/stable-linux-4.4.y/scripts/Makefile.clean obj=arch/x86/entry
make -f /home/inaddy/work/sources/linux/stable/stable-linux-4.4.y/scripts/Makefile.clean obj=arch/x86/ia32
make -f /home/inaddy/work/sources/linux/stable/stable-linux-4.4.y/scripts/Makefile.clean obj=arch/x86/kernel
```

Debugging: Real Cases
BUG #3303 (Provisioning the tools)

```
inaddy@workstation:~$ lxc-ls -f
NAME          STATE   AUTOSTART GROUPS IPV4          IPV6
worklxcamd64 STOPPED 0         -      -             -
worklxci686  STOPPED 0         -      -             -

$ lxc-copy -B overlay -s -n worklxcamd64 -N bug3303

inaddy@workstation:~$ lxc-ls -f
NAME          STATE   AUTOSTART GROUPS IPV4          IPV6
bug3303       STOPPED 0         -      -             -
worklxcamd64 STOPPED 0         -      -             -
worklxci686  STOPPED 0         -      -             -

inaddy@workstation:~$ lxc-start -n bug3303

inaddy@workstation:~$ ssh bug3303

(c)inaddy@bug3303:~$ uname -a
Linux bug3303 4.16.0-2-amd64 #1 SMP Debian 4.16.16-2
(2018-06-22) x86_64 GNU/Linux

(c)inaddy@bug3303:~$ lsb_release -a
No LSB modules are available.
Distributor ID:     Debian
Description:  Debian GNU/Linux unstable (sid)
Release:      unstable
Codename:     sid
```

```
(c)inaddy@bug3303:~$ ls
bug3303lxc  cron.sh  tools  work

(c)inaddy@bug3303:~$ cd work
(c)inaddy@bug3303:~/work$ ls
0_update_all.sh  files    notes  README.md  scripts
build            kernels  pkgs   scratch    sources

(c)inaddy@bug3303:~$ cd work/kernels
(c)inaddy@bug3303:~/work/kernels$ ls
amd64  arm64  armhf  i386

(c)inaddy@bug3303:~/work/kernels$ cd amd64
(c)inaddy@bug3303:~/work/kernels/amd64$ ls
mainline  stable  stable-rc

(c)inaddy@bug3303:~$ cd stable/stable-linux-4.4.y
(c)inaddy@bug3303:~/../stable-linux-4.4.y$ ls
linux-firmware-image-4.4.154_4.4.154-1_amd64.deb
linux-headers-4.4.154_4.4.154-1_amd64.deb
linux-image-4.4.154_4.4.154-1_amd64.deb
linux-image-4.4.154-dbg_4.4.154-1_amd64.deb
linux-libc-dev_4.4.154-1_amd64.deb
```

Debugging: Real Cases
BUG #3303 (Install generated .deb packages)

```
(c)inaddy@bug3303:~/work/kernels/amd64/stable/stable-linux-4.4.$ sudo dpkg -i ./*.deb

Selecting previously unselected package linux-firmware-image-4.4.154.
(Reading database ... 67913 files and directories currently installed.)
Preparing to unpack .../linux-firmware-image-4.4.154_4.4.154-1_amd64.deb ...
Unpacking linux-firmware-image-4.4.154 (4.4.154-1) ...
Selecting previously unselected package linux-headers-4.4.154.
Preparing to unpack .../linux-headers-4.4.154_4.4.154-1_amd64.deb ...
Unpacking linux-headers-4.4.154 (4.4.154-1) ...
Selecting previously unselected package linux-image-4.4.154.
Preparing to unpack .../linux-image-4.4.154_4.4.154-1_amd64.deb ...
Unpacking linux-image-4.4.154 (4.4.154-1) ...
Selecting previously unselected package linux-image-4.4.154-dbg.
Preparing to unpack .../linux-image-4.4.154-dbg_4.4.154-1_amd64.deb ...
Unpacking linux-image-4.4.154-dbg (4.4.154-1) ...
dpkg: warning: downgrading linux-libc-dev:amd64 from 4.18.6-1 to 4.4.154-1
Preparing to unpack .../linux-libc-dev_4.4.154-1_amd64.deb ...
Unpacking linux-libc-dev (4.4.154-1) over (4.18.6-1) ...
dpkg: warning: unable to delete old directory '/usr/include/x86_64-linux-gnu/asm': Directory not empty
Setting up linux-firmware-image-4.4.154 (4.4.154-1) ...
Setting up linux-headers-4.4.154 (4.4.154-1) ...
Setting up linux-image-4.4.154 (4.4.154-1) ...
update-initramfs: Generating /boot/initrd.img-4.4.154
Setting up linux-image-4.4.154-dbg (4.4.154-1) ...
Setting up linux-libc-dev (4.4.154-1) ...
```

Debugging: Real Cases
BUG #3303 (Install the tools - optional)

```
(c)inaddy@bug3303:~/work/kernels/amd64/stable/stable-linux-4.4.y$ apt-get install crash

Reading package lists... Done
Building dependency tree
Reading state information... Done
...
The following additional packages will be installed:
  libsnappy1v5
The following NEW packages will be installed:
  crash libsnappy1v5
0 upgraded, 2 newly installed, 0 to remove and 1 not upgraded.
Need to get 2,858 kB of archives.
After this operation, 8,765 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian sid/main amd64 libsnappy1v5 amd64 1.1.7-1 [17.0 kB]
Get:2 http://deb.debian.org/debian sid/main amd64 crash amd64 7.2.3+real-1 [2,841 kB]
Fetched 2,858 kB in 1s (5,486 kB/s)
...
Unpacking crash (7.2.3+real-1) ...
Processing triggers for libc-bin (2.27-6) ...
Setting up libsnappy1v5:amd64 (1.1.7-1) ...
Processing triggers for man-db (2.8.4-2) ...
Setting up crash (7.2.3+real-1) ...
Processing triggers for libc-bin (2.27-6) ...
```

Debugging: Real Cases
BUG #3303 (Provision the reproducer)

```
inaddy@workstation:~$ virsh list --all
 Id    Name                           State
----------------------------------------------------
 -     workkvmamd64                   shut off
 -     workkvmamd64stretch            shut off
 -     workkvmi686                    shut off
 -     workqemuamd64                  shut off
 -     workqemuarm64                  shut off
 -     workqemuarmhf                  shut off
 -     workqemui686                   shut off

inaddy@workstation:~$ virtclone.sh workkvmamd64 bug3303kvm

running:
- qcowhostname.sh bug3303kvm
bug3303kvm
- qcowhome.sh bug3303kvm
sending home files to lxc8599 (bug3303kvm)...

$ ls ~/work/kernels/amd64/stable/stable-linux-4.4.y

linux-firmware-image-4.4.154_4.4.154-1_amd64.deb
linux-headers-4.4.154_4.4.154-1_amd64.deb
linux-image-4.4.154_4.4.154-1_amd64.deb
linux-image-4.4.154-dbg_4.4.154-1_amd64.deb
linux-libc-dev_4.4.154-1_amd64.deb
```

```
$ qcowkerninst.sh bug3303kvm

Selecting previously unselected package linux-firmware-image-4.4.154.
(Reading database ... 87049 files and directories currently
installed.)
Preparing to unpack
.../linux-firmware-image-4.4.154_4.4.154-1_amd64.deb ...
Unpacking linux-firmware-image-4.4.154 (4.4.154-1) ...
Setting up linux-firmware-image-4.4.154 (4.4.154-1) ...
Selecting previously unselected package linux-headers-4.4.154.
(Reading database ... 87200 files and directories currently
installed.)
Preparing to unpack .../linux-headers-4.4.154_4.4.154-1_amd64.deb ...
Unpacking linux-headers-4.4.154 (4.4.154-1) ...
Setting up linux-headers-4.4.154 (4.4.154-1) ...
Selecting previously unselected package linux-image-4.4.154.
(Reading database ... 104731 files and directories currently
installed.)
Preparing to unpack .../linux-image-4.4.154_4.4.154-1_amd64.deb ...
Unpacking linux-image-4.4.154 (4.4.154-1) ...
Setting up linux-image-4.4.154 (4.4.154-1) ...
update-initramfs: Generating /boot/initrd.img-4.4.154
kdump-tools: Generating /var/lib/kdump/initrd.img-4.4.154
(Reading database ... 108443 files and directories currently
installed.)
Preparing to unpack .../linux-libc-dev_4.4.154-1_amd64.deb ...
Unpacking linux-libc-dev (4.4.154-1) over (4.18.6-1) ...
dpkg: warning: unable to delete old directory
'/usr/include/x86_64-linux-gnu/asm': Directory not empty
Setting up linux-libc-dev (4.4.154-1) ...
```

Debugging: Real Cases
BUG #3303 (Reproduce using QEMU or Boards)

```
inaddy@workstation:~$ qcowvmlinuz.sh bug3303kvm 4.4.154
bringing lxc28566 (bug3303kvm) kernel/ramdisk to host

inaddy@workstation:~$ virsh start bug3303kvm
Domain bug3303kvm started

inaddy@workstation:~$ ping bug3303kvm
PING bug3303kvm.celeiro.br (192.168.49.25) 56(84) bytes of data.
64 bytes from bug3303kvm.celeiro.br (192.168.49.25): icmp_seq=1 ttl=64 time=0.172 ms

inaddy@workstation:~$ ssh bug3303

(c)inaddy@bug3303:~$ uname -a
Linux bug3303 4.16.0-2-amd64 #1 SMP Debian 4.16.16-2 (2018-06-22) x86_64 GNU/Linux

(k)inaddy@bug3303kvm:~$ git clone git@github.com:rafaeldtinoco/work.git
Cloning into 'work'...
remote: Counting objects: 1240, done.
remote: Compressing objects: 100% (119/119), done.
remote: Total 1240 (delta 128), reused 186 (delta 102), pack-reused 1017
Receiving objects: 100% (1240/1240), 23.30 MiB | 5.98 MiB/s, done.
Resolving deltas: 100% (719/719), done.

(k)inaddy@bug3303kvm:~$ ls ~/work/scripts/
alienize.sh        build.sh          project.sh       qcowkerninst.sh  update.sh
bug_readme.sh      create_branch.sh  qcowcmd.sh       qcowshell.sh     virtclone.sh
builddeb.sh        create.sh         qcowhome.sh      qcowvmlinuz.sh   virtdel.sh
buildkselftest.sh  gettestpkg.sh     qcowhostname.sh  update_all.sh    weblatest.sh
```

Debugging: Real Cases
BUG #3303 (Make sure KDUMP is configured)

```
(k)inaddy@bug3303kvm:~$ cat /proc/cmdline

root=/dev/vda noresume console=tty0 console=ttyS0,38400n8 apparmor=0 net.ifnames=0
crashkernel=256M

(k)inaddy@bug3303kvm:~$ sudo kdump-config status

current state    : ready to kdump

(k)inaddy@bug3303kvm:~$ dmesg | grep -i reserv

...
[    0.000000] Reserving 256MB of memory at 640MB for crashkernel (System RAM: 4095MB)
[    0.000000]   DMA zone: 22 pages reserved
[    0.000000] Memory: 3766304K/4193768K available (6735K kernel code, 1182K rwdata, 3300K
rodata, 1388K init, 724K bss, 427464K reserved, 0K cma-reserved)
…

(k)inaddy@bug3303kvm:~$ sudo cat /proc/iomem | grep -i crash

  28000000-37ffffff : Crash kernel
```

Debugging: Real Cases
BUG #3303 (Manual or Automated KDUMP ?)

```
(k)inaddy@bug3303kvm:~$  sysctl -a 2>&1 | grep panic

kernel.hardlockup_panic = 1
kernel.hung_task_panic = 1
kernel.panic = 1
kernel.panic_on_io_nmi = 1
kernel.panic_on_oops = 1
kernel.panic_on_unrecovered_nmi = 1
kernel.panic_on_warn = 1
kernel.softlockup_panic = 1
kernel.unknown_nmi_panic = 1
vm.panic_on_oom = 1
```

Debugging: Real Cases
BUG #3303 (Use "gettestpkg.sh" - optional)

```
(k)inaddy@bug3303kvm:~$ gettestpkg.sh

usage: /home/inaddy/work/scripts/gettestpkg.sh -h | [1] [2] [3] [4] [5]

[1] = get | install
[2] = kselftest | ltp | libhugetlbfs
[3] = armhf | arm64 | i386 | amd64
[4] = rpm | deb | txz
[5] = args

[5] (kselftest): stable-4.14 | stable-4.17 stable-4.18 mainline next

examples:

/home/inaddy/work/scripts/gettestpkg.sh get ltp armhf deb
/home/inaddy/work/scripts/gettestpkg.sh get libhugetlb amd64 rpm
/home/inaddy/work/scripts/gettestpkg.sh get kselftest i386 txz stable-4.18


(k)inaddy@bug3303kvm:~$ cat .aliases | grep ltps -A4

ltps() {
    TEST=$1
    [ ! $TEST ] && return
    (cd /opt/ltp ; sudo ./runltp -s $TEST -p -q -d /tmp -i 2 -I 2 ; cd -)
}
```

Debugging: Real Cases
BUG #3303 (Install the test suite package)

```
(k)inaddy@bug3303kvm:~$  gettestpkg.sh get ltp amd64 deb

downloading ltp_20180515-258-g37fa79858_amd64.deb... complete!

(k)inaddy@bug3303kvm:~$  sudo dpkg -i ./ltp_20180515-258-g37fa79858_amd64.deb

Selecting previously unselected package ltp:amd64.
(Reading database ... 108539 files and directories currently installed.)
Preparing to unpack .../ltp_20180515-258-g37fa79858_amd64.deb ...
Unpacking ltp:amd64 (20180515-258-g37fa79858) ...
Setting up ltp:amd64 (20180515-258-g37fa79858) ...
```

Debugging: Real Cases
BUG #3303 (Finally: Reproduce it)

```
(k)inaddy@bug3303kvm:~$ ltps fanotify07
...
Running tests.......
tst_test.c:1063: INFO: Timeout per run is 0h 05m 00s
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Cannot kill test processes!
Congratulation, likely test hit a kernel bug.
Exitting uncleanly...
...
```

```
tst_test.c:1063: INFO: Timeout per run is 0h 05m 00s
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Test timeouted, sending SIGKILL!
Cannot kill test processes!
Congratulation, likely test hit a kernel bug.
Exitting uncleanly...
INFO: ltp-pan reported some tests FAIL
LTP Version: 20180515-255-ge05f958d2
INFO: Test end time: Sun Sep  9 01:45:23 UTC 2018
```

Debugging: Real Cases
BUG #3303 (... without automated panic)

```
(k)inaddy@bug3303kvm:~$ ps -ef | grep fano
root      1385     1  0 Sep09 pts/0    00:00:00 fanotify07
root      1386  1385  0 Sep09 pts/0    00:00:00 fanotify07
root      1387  1385  0 Sep09 pts/0    00:00:00 fanotify07
root      1388  1385  0 Sep09 pts/0    00:00:00 fanotify07
root      1389  1385  0 Sep09 pts/0    00:00:00 fanotify07
root      1390  1385  0 Sep09 pts/0    00:00:00 fanotify07
root      1391  1385  0 Sep09 pts/0    00:00:00 fanotify07
root      1392  1385  0 Sep09 pts/0    00:00:00 fanotify07
root      1393  1385  0 Sep09 pts/0    00:00:00 fanotify07
root      1394  1385  0 Sep09 pts/0    00:00:00 fanotify07
root      1395  1385  0 Sep09 pts/0    00:00:00 fanotify07
root      1396  1385  0 Sep09 pts/0    00:00:00 fanotify07
root      1397  1385  0 Sep09 pts/0    00:00:00 fanotify07
root      1398  1385  0 Sep09 pts/0    00:00:00 fanotify07
root      1399  1385  0 Sep09 pts/0    00:00:00 fanotify07
root      1400  1385  0 Sep09 pts/0    00:00:00 fanotify07
root      1401  1385  0 Sep09 pts/0    00:00:00 fanotify07
root      1507     1  0 Sep09 pts/0    00:00:00 fanotify07
root      1508  1507  0 Sep09 pts/0    00:00:00 fanotify07
root      1509  1507  0 Sep09 pts/0    00:00:00 fanotify07
root      1510  1507  0 Sep09 pts/0    00:00:00 fanotify07
root      1511  1507  0 Sep09 pts/0    00:00:00 fanotify07
root      1512  1507  0 Sep09 pts/0    00:00:00 fanotify07
root      1513  1507  0 Sep09 pts/0    00:00:00 fanotify07
root      1514  1507  0 Sep09 pts/0    00:00:00 fanotify07
root      1515  1507  0 Sep09 pts/0    00:00:00 fanotify07
root      1516  1507  0 Sep09 pts/0    00:00:00 fanotify07
root      1517  1507  0 Sep09 pts/0    00:00:00 fanotify07
root      1518  1507  0 Sep09 pts/0    00:00:00 fanotify07
root      1519  1507  0 Sep09 pts/0    00:00:00 fanotify07
root      1520  1507  0 Sep09 pts/0    00:00:00 fanotify07
root      1521  1507  0 Sep09 pts/0    00:00:00 fanotify07
root      1522  1507  0 Sep09 pts/0    00:00:00 fanotify07
root      1523  1507  0 Sep09 pts/0    00:00:00 fanotify07

(k)inaddy@bug3303kvm:~$ sudo cat /proc/1507/stack
[<ffffffff810de59f>] __synchronize_srcu+0x10f/0x140
[<ffffffff810de5f4>] synchronize_srcu+0x24/0x30
[<ffffffff8123681b>] fsnotify_destroy_group+0x3b/0x70
[<ffffffff812399b0>] fanotify_release+0x100/0x150
[<ffffffff811f4907>] __fput+0xa7/0x210
[<ffffffff811f4aae>] ____fput+0xe/0x10
[<ffffffff8109db7f>] task_work_run+0x7f/0xa0
[<ffffffff81003233>] exit_to_usermode_loop+0xa3/0xb0
[<ffffffff81003c6e>] syscall_return_slowpath+0x4e/0x50
[<ffffffff8168ef2b>] int_ret_from_sys_call+0x25/0x93
[<ffffffffffffffff>] 0xffffffffffffffff
```

... with automated

PANIC

Debugging: Real Cases
BUG #3303 (KDUMP generated ?)

```
(k)inaddy@bug3303kvm:~$ sudo su -

(k)root@bug3303kvm:~$ cd /var/crash
(k)root@bug3303kvm:/var/crash$ cd 201809102030
(k)root@bug3303kvm:/var/crash/201809102030$ ls

dmesg.201809102030  dump.201809102030

(k)root@bug3303kvm:/var/crash/201809102030$ tail -20 dmesg.201809102030

[  360.403327] Kernel panic - not syncing: hung_task: blocked tasks
[  360.404812] CPU: 2 PID: 29 Comm: khungtaskd Not tainted 4.4.154 #1
[  360.406716] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.11.1-1 04/01/2014
[  360.407319]  0000000000000086 2168444805dbcf8a ffff88013a523de0 ffffffff8138cc16
[  360.407319]  ffffffff81a50284 ffff88013a55e200 ffff88013a523e60 ffffffff81171c8b
[  360.407319]  0000000000000008 ffff88013a523e70 ffff88013a523e10 2168444805dbcf8a
[  360.407319] Call Trace:
[  360.407319]  [<ffffffff8138cc16>] dump_stack+0x63/0x8d
[  360.407319]  [<ffffffff81171c8b>] panic+0xd6/0x22a
[  360.407319]  [<ffffffff81391389>] ? nmi_trigger_all_cpu_backtrace+0x1e9/0x271
[  360.407319]  [<ffffffff811788a0>] watchdog.cold.2+0xa4/0xa4
[  360.407319]  [<ffffffff81127490>] ? reset_hung_task_detector+0x20/0x20
[  360.407319]  [<ffffffff8109f617>] kthread+0xe7/0x100
[  360.407319]  [<ffffffff8109f530>] ? kthread_create_on_node+0x1a0/0x1a0
[  360.407319]  [<ffffffff8168f1a2>] ret_from_fork+0x42/0x80
[  360.407319]  [<ffffffff8109f530>] ? kthread_create_on_node+0x1a0/0x1a0
```

Debugging: Real Cases
BUG #3303 (Open it using crash)

```
(c)root@bug3303:~$ cd ~inaddy/work/scratch/201809102030

$ crash /usr/lib/debug/boot/vmlinux-4.4.154 ./dump.201809102030
…
       KERNEL: /usr/lib/debug/boot/vmlinux-4.4.154
     DUMPFILE: ./dump.201809102030  [PARTIAL DUMP]
         CPUS: 4
         DATE: Mon Sep 10 20:30:28 2018
       UPTIME: 00:06:00
LOAD AVERAGE: 18.66, 8.14, 3.09
        TASKS: 141
     NODENAME: bug3303kvm
      RELEASE: 4.4.154
      VERSION: #1 SMP Thu Sep 6 11:45:05 -03 2018
      MACHINE: x86_64  (4013 Mhz)
       MEMORY: 4 GB
        PANIC: "Kernel panic - not syncing: hung_task: blocked tasks"
          PID: 29
      COMMAND: "khungtaskd"
         TASK: ffff88013a45b800  [THREAD_INFO: ffff88013a520000]
          CPU: 2
        STATE: TASK_RUNNING (PANIC)

crash> dmesg
```

Debugging: Real Cases
BUG #3303 (kdump dmesg: Obvious 1st move)

```
[  360.148021] INFO: task fsnotify_mark:41 blocked for more than 120 seconds.

...
[  360.161700] Call Trace:
[  360.162314]  [<ffffffff8168adbb>] schedule+0x2b/0x80
[  360.163465]  [<ffffffff8168de32>] schedule_timeout+0x252/0x2a0
[  360.164808]  [<ffffffff8168a848>] ? __schedule+0x2a8/0x7f0
[  360.169854]  [<ffffffff8168a848>] ? __schedule+0x2a8/0x7f0
[  360.171091]  [<ffffffff810c7981>] ?  _raw_callee_save__pv_queued_spin_unlock+0x11/0x2
[  360.172879]  [<ffffffff8168c299>] wait_for_completion+0xf9/0x140
[  360.174238]  [<ffffffff810aaba0>] ? wake_up_q+0x70/0x70
[  360.175446]  [<ffffffff810de59f>] __synchronize_srcu+0x10f/0x140
[  360.176839]  [<ffffffff810dda60>] ? trace_raw_output_rcu_utilization+0x60/0x60
[  360.178454]  [<ffffffff810de5f4>] synchronize_srcu+0x24/0x30
[  360.179755]  [<ffffffff81236e05>] fsnotify_mark_destroy+0x85/0x130
[  360.181197]  [<ffffffff810c2510>] ? prepare_to_wait_event+0xd0/0xd0
[  360.182639]  [<ffffffff81236d80>] ? fsnotify_put_mark+0x40/0x40
[  360.183968]  [<ffffffff8109f617>] kthread+0xe7/0x100
[  360.185133]  [<ffffffff8109f530>] ? kthread_create_on_node+0x1a0/0x1a0
[  360.186767]  [<ffffffff8168f1a2>] ret_from_fork+0x42/0x80
[  360.188402]  [<ffffffff8109f530>] ? kthread_create_on_node+0x1a0/0x1a0


[  360.190196] Sending NMI to all CPUs:
...
```

Debugging: Real Cases
BUG #3303 (backtrace won't always help)

```
Sending NMI to all CPUs:
NMI backtrace for cpu 0
CPU: 0 PID: 182 Comm: jbd2/vda-8 Not tainted 4.4.154 #1
Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), ...
task: ffff8800b8e80000 ti: ffff88007f86c000 task.ti: ffff88007f86c000
RIP: 0010:[<...>]  [<ffffffff810604fa>] native_write_msr_safe
RSP: 0018:ffff88007f86fb48  EFLAGS: 00000016
RAX: 00000000000000fd RBX: 0000000000000003 RCX: 0000000000000830
RDX: 0000000000000003 RSI: 00000000000000fd RDI: 0000000000000830
RBP: ffff88007f86fb48 R08: ffffffffffffffff R09: 0000000000000008
R10: 0000000000002c00 R11: ffff8800b8e80060 R12: ffffffff81811b40
R13: 0000000000080000 R14: 0000000000000000 R15: 00000000000000fd
FS:  00007fc6c4db3740(0000) GS:ffff88013fc00000(0000)...
CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
CR2: 000055971858c410 CR3: 00000000bb630000 CR4: 0000000000406f0
Stack:
 ffff88007f86fb88 ffffffff81056cdc 0000000000000092 ffff88013fd80000
 ffff8800bb7714c4 0000000000000000 ffff8800bb770e00 0000000000016d00
 ffff88007f86fb98 ffffffff81056d73 ffff88007f86fba8 ffffffff8104e554
Call Trace:
 [<ffffffff81056cdc>] __x2apic_send_IPI_mask+0xac/0xd0
 [<ffffffff81056d73>] x2apic_send_IPI_mask+0x13/0x20
 [<ffffffff8104e554>] native_smp_send_reschedule+0x54/0x70
 [<ffffffff810aa9d7>] try_to_wake_up+0x287/0x3c0
 [<ffffffff810aabb2>] default_wake_function+0x12/0x20
 [<ffffffff810c2528>] autoremove_wake_function+0x18/0x50
 [<ffffffff810c1df3>] __wake_up_common+0x53/0x90
 [<ffffffff810c1e69>] __wake_up+0x39/0x50
 [<ffffffff812d3fb3>] jbd2_journal_commit_transaction+0x15f3/0x17d0
 [<ffffffff810b317f>] ? account_entity_dequeue+0xaf/0xd0
 [<ffffffff812d7b29>] kjournald2+0xc9/0x270
 [<ffffffff810c2510>] ? prepare_to_wait_event+0xd0/0xd0
 [<ffffffff812d7a60>] ? commit_timeout+0x10/0x10
 [<ffffffff8109f617>] kthread+0xe7/0x100
 [<ffffffff8109f530>] ? kthread_create_on_node+0x1a0/0x1a0
 [<ffffffff8168f1a2>] ret_from_fork+0x42/0x80
 [<ffffffff8109f530>] ? kthread_create_on_node+0x1a0/0x1a0
Code: 00 55 89 f9 48 89 e5 0f 32 31 ff 48 c1 e2 20 89 3e 5d 48 09 d0 c3
66
NMI backtrace for cpu 1
CPU: 1 PID: 0 Comm: swapper/1 Not tainted 4.4.154 #1
Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), ...
task: ffff88013abc0e00 ti: ffff88013abc8000 task.ti: ffff88013abc8000
RIP: 0010:[<...>]  [<ffffffff81060606>] native_safe_halt+0x6/
RSP: 0018:ffff88013abcbea0  EFLAGS: 00000246
RAX: ffffffff81020e10 RBX: 0000000000000000 RCX: 0000000000000000
RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000000
RBP: ffff88013abcbea0 R08: 0000000000000000 R09: 0000000000000000
R10: ffff88013901ac00 R11: 0000000100003ac8 R12: 0000000000000001
R13: 00000000ffffffff R14: 0000000000000000 R15: 0000000000000000
FS:  00007f2b2818e940(0000) GS:ffff88013fc80000(0000)
knlGS:00000000000000
CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
CR2: 00007f2b26125000 CR3: 000000013a63e000 CR4: 0000000000406f0
Stack:
 ffff88013abcbec8 ffffffff81020e30 0000000000000000 ffffffff81d1ea40
 00000000ffffffff ffff88013abcbed8 ffffffff810219b5 ffff88013abcbee8
 ffffffff810c2873 ffff88013abcbf28 ffffffff810c2bf4 23d31290a9cc5810
Call Trace:
 [<ffffffff81020e30>] default_idle+0x20/0xf0
 [<ffffffff810219b5>] arch_cpu_idle+0x15/0x20
 [<ffffffff810c2873>] default_idle_call+0x33/0x40
 [<ffffffff810c2bf4>] cpu_startup_entry+0x314/0x350
 [<ffffffff8104f1a7>] start_secondary+0x177/0x1b0
```

Debugging: Real Cases
BUG #3303 (is the trace an effect from PANIC ?)

```
NMI backtrace for cpu 2
CPU: 2 PID: 29 Comm: khungtaskd Not tainted 4.4.154 #1
Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), ...
task: ffff88013a45b800 ti: ffff88013a520000 task.ti: ffff88013a520000
RIP: 0010:[<...>]  [<ffffffff810604fa>] native_write_msr_safe
RSP: 0018:ffff88013a523da8  EFLAGS: 00000012
RAX: 0000000000000400 RBX: 0000000000000002 RCX: 0000000000000830
RDX: 0000000000000002 RSI: 0000000000000400 RDI: 0000000000000830
RBP: ffff88013a523da8 R08: fffffffffffffffc R09: 000000000000000f
R10: ffff8800000b8f00 R11: ffffffff813fab30 R12: ffffffff81d22720
R13: 0000000000080000 R14: 0000000000000002 R15: 0000000000000400
FS:  00007f9acf4d4740(0000) GS:ffff88013fd00000(0000) knlGS:...
CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
CR2: 00007fe957bf1bb0 CR3: 00000000bb20a000 CR4: 00000000000406f0
Stack:
 ffff88013a523de8 ffffffff81056cdc 0000000000000296 0000000000013980
 0000000000000001 ffffffff810537c0 00000000003fffd9 00000000000003d9
 ffff88013a523df8 ffffffff81056d73 ffff88013a523e08 ffffffff810537e1
Call Trace:
 [<ffffffff81056cdc>] __x2apic_send_IPI_mask+0xac/0xd0
 [<ffffffff810537c0>] ? irq_force_complete_move+0x150/0x150
 [<ffffffff81056d73>] x2apic_send_IPI_mask+0x13/0x20
 [<ffffffff810537e1>] nmi_raise_cpu_backtrace+0x21/0x30
 [<ffffffff813914ba>] nmi_trigger_all_cpu_backtrace.cold.4+0x57/0x8d
 [<ffffffff81053849>] arch_trigger_all_cpu_backtrace+0x19/0x20
 [<ffffffff81178894>] watchdog.cold.2+0x98/0xa4
 [<ffffffff81127490>] ? reset_hung_task_detector+0x20/0x20
 [<ffffffff8109f617>] kthread+0xe7/0x100
 [<ffffffff8109f530>] ? kthread_create_on_node+0x1a0/0x1a0
 [<ffffffff8168f1a2>] ret_from_fork+0x42/0x80
 [<ffffffff8109f530>] ? kthread_create_on_node+0x1a0/0x1a0
Code: 00 55 89 f9 48 89 e5 0f 32 31 ff 48 c1 e2 20 89 3e 5d 48 09 d0 c3
66
NMI backtrace for cpu 3
CPU: 3 PID: 1343 Comm: genload Not tainted 4.4.154 #1
Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.11.1-1
04/01
task: ffff8800bb770e00 ti: ffff88007f860000 task.ti: ffff88007f860000
RIP: 0010:[<ffffffff8168eae4>]  [<...>] _raw_spin_lock+0x14/0
RSP: 0018:ffff88007f863ee8  EFLAGS: 00000246
RAX: 0000000000000000 RBX: ffff88013ac28e78 RCX: ffff88013ac28f10
RDX: 0000000000000001 RSI: 0000000000000000 RDI: ffff88013ac28e78
RBP: ffff88007f863ee8 R08: ffff88007f472890 R09: 0000000000000000
R10: 00000053dce0b700 R11: 0000000000321b97 R12: ffff88013ac28f58
R13: 0000000000000000 R14: ffffffff81226f90 R15: ffff88013ac28df0
FS:  00007f9acf4d4740(0000) GS:ffff88013fd80000(0000)
knlGS:00000000000000
CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
CR2: 00007f89c0d4c9cf CR3: 000000007f42c000 CR4: 00000000000406f0
Stack:
 ffff88007f863f28 ffffffff812300d9 0000000000000000 0000000000000000
 0000000000000002 0000000000001770 0000000000000001 0000000000000000
 ffff88007f863f48 ffffffff81227302 0000000100000000 aa0f8fbe9643fe6c
Call Trace:
 [<ffffffff812300d9>] iterate_bdevs+0x109/0x160
 [<ffffffff81227302>] sys_sync+0x72/0xb0
 [<ffffffff8168eda5>] entry_SYSCALL_64_fastpath+0x22/0x99
Code: c3 e8 61 a5 a3 ff 5d c3 66 66 2e 0f 1f 84 00 00 00 00 00 0f 1f 40
00
```

Debugging: Real Cases
BUG #3303 (Trace from all affected tasks)

```
crash> foreach fanotify07 bt

PID: 1451   TASK: ffff8800b8c4e200  CPU: 2   COMMAND: "fanotify07"
 #0 [ffff8800bb2efe20] __schedule at ffffffff8168a825
 #1 [ffff8800bb2efe70] schedule at ffffffff8168adbb
 #2 [ffff8800bb2efe80] do_wait at ffffffff8108356f
 #3 [ffff8800bb2efec0] sys_wait4 at ffffffff810845e4
 #4 [ffff8800bb2eff50] entry_SYSCALL_64_fastpath at ffffffff8168eda5
    RIP: 00007f89c0dfe1c7  RSP: 00007ffedf553f98  RFLAGS: 00000246
    RAX: fffffffffffffffda  RBX: 0000000000000000  RCX:
00007f89c0dfe1c7
    RDX: 0000000000000000  RSI: 00007ffedf553ff0  RDI:
00000000000005ac
    RBP: 00000000000005ac   R8: 00007ffedf553ff0   R9:
0000000000000000
    R10: 0000000000000000  R11: 0000000000000246  R12:
00007ffedf553ff0
    R13: 0000563a4ea799ea  R14: 0000000000000446  R15:
0000000000000000
    ORIG_RAX: 000000000000003d  CS: 0033  SS: 002b

PID: 1452   TASK: ffff8800b8c4aa00  CPU: 0   COMMAND: "fanotify07"
 #0 [ffff8800b8c47c48] __schedule at ffffffff8168a825
 #1 [ffff8800b8c47c98] schedule at ffffffff8168adbb
 #2 [ffff8800b8c47ca8] schedule_timeout at ffffffff8168de32
 #3 [ffff8800b8c47d30] wait_for_completion at ffffffff8168c299
 #4 [ffff8800b8c47d88] __synchronize_srcu at ffffffff810de59f
 #5 [ffff8800b8c47df8] synchronize_srcu at ffffffff810de5f4
 #6 [ffff8800b8c47e10] fsnotify_destroy_group at ffffffff8123681b
 #7 [ffff8800b8c47e30] fanotify_release at ffffffff812399b0
 #8 [ffff8800b8c47e78] __fput at ffffffff811f4907
    RIP: 00007f89c0e22a24  RSP: 00007ffedf553f68  RFLAGS: 00000246
    RAX: 0000000000000000  RBX: 0000563a4ea826a0  RCX: 00007f89c0e22a24
    RDX: 0000000000020000  RSI: 0000000000000001  RDI: 0000000000000007
    RBP: 0000563a4ea826e0   R8: 0000563a4ea82800   R9: 00007f89c0ef0240
    R10: 00000000ffffff9c  R11: 0000000000000246  R12: 0000000000000007
    R13: 0000000000000004  R14: 00007ffedf553f90  R15: a3d70a3d70a3d70b
    ORIG_RAX: 0000000000000003  CS: 0033  SS: 002b

PID: 1453   TASK: ffff8800b8e58e00  CPU: 2   COMMAND: "fanotify07"
 #0 [ffff8800bb1cbcd8] __schedule at ffffffff8168a825
 #1 [ffff8800bb1cbd28] schedule at ffffffff8168adbb
 #2 [ffff8800bb1cbd38] fanotify_handle_event at ffffffff812396a3
 #3 [ffff8800bb1cbdb8] fsnotify at ffffffff81235ea9
 #4 [ffff8800bb1cbe88] security_file_permission at ffffffff812fc086
 #5 [ffff8800bb1cbeb8] rw_verify_area at ffffffff811f2ccf
 #6 [ffff8800bb1cbed8] vfs_read at ffffffff811f2dd6
 #7 [ffff8800bb1cbf10] sys_read at ffffffff811f3b85
 #8 [ffff8800bb1cbf50] entry_SYSCALL_64_fastpath at ffffffff8168eda5
    RIP: 00007f89c0e221d1  RSP: 00007ffedf553f68  RFLAGS: 00000246
    RAX: fffffffffffffffda  RBX: 00007f89c0d425f8  RCX: 00007f89c0e221d1
    RDX: 0000000000000100  RSI: 0000563a4ea82700  RDI: 0000000000000006
    RBP: 00007f89c0f07560   R8: 00007f89c0f08540   R9: 00007f89c0f08540
    R10: 000000000000039c  R11: 0000000000000246  R12: 00007f89c0f07000
    R13: 0000000000000001  R14: 000000007c9d4d41  R15: 00007ffedf553a44
    ORIG_RAX: 0000000000000000  CS: 0033  SS: 002b

PID: 1454   TASK: ffff8800b8e5d400  CPU: 1   COMMAND: "fanotify07"
 #0 [ffff8800bb1cfcd8] __schedule at ffffffff8168a825
 …

THIS IS INTENSE WORK =) … I'LL HELP YOU A BIT!
```

Debugging: Real Cases
BUG #3303 (Our case: 3 stacktrace patterns)

```
PID: 1451   TASK: ffff8800b8c4e200  CPU: 2   COMMAND: "fanotify07"

 #0 [ffff8800bb2efe20] __schedule at ffffffff8168a825
 #1 [ffff8800bb2efe70] schedule at ffffffff8168adbb
 #2 [ffff8800bb2efe80] do_wait at ffffffff8108356f
 #3 [ffff8800bb2efec0] sys_wait4 at ffffffff810845e4
 #4 [ffff8800bb2eff50] entry_SYSCALL_64_fastpath at ffffffff8168eda5
```

```
PID: 1452   TASK: ffff8800b8c4aa00  CPU: 0   COMMAND: "fanotify07"

 #0 [ffff8800b8c47c48] __schedule at ffffffff8168a825
 #1 [ffff8800b8c47c98] schedule at ffffffff8168adbb
 #2 [ffff8800b8c47ca8] schedule_timeout at ffffffff8168de32
 #3 [ffff8800b8c47d30] wait_for_completion at ffffffff8168c299
 #4 [ffff8800b8c47d88] __synchronize_srcu at ffffffff810de59f
 #5 [ffff8800b8c47df8] synchronize_srcu at ffffffff810de5f4
 #6 [ffff8800b8c47e10] fsnotify_destroy_group at ffffffff8123681b
 #7 [ffff8800b8c47e30] fanotify_release at ffffffff812399b0
 #8 [ffff8800b8c47e78] __fput at ffffffff811f4907
 #9 [ffff8800b8c47ec0] ____fput at ffffffff811f4aae
#10 [ffff8800b8c47ed0] task_work_run at ffffffff8109db7f
#11 [ffff8800b8c47f08] exit_to_usermode_loop at ffffffff81003233
#12 [ffff8800b8c47f30] syscall_return_slowpath at ffffffff81003c6e
#13 [ffff8800b8c47f50] int_ret_from_sys_call at ffffffff8168ef2b
```

```
PID: 1453   TASK: ffff8800b8e58e00  CPU: 2   COMMAND: "fanotify07"
PID: 1454   TASK: ffff8800b8e5d400  CPU: 1   COMMAND: "fanotify07"
PID: 1455   TASK: ffff8800b8e5b800  CPU: 2   COMMAND: "fanotify07"
PID: 1456   TASK: ffff8800b8e59c00  CPU: 2   COMMAND: "fanotify07"
PID: 1457   TASK: ffff8800b8e5c600  CPU: 2   COMMAND: "fanotify07"
PID: 1458   TASK: ffff8800b8e5e200  CPU: 2   COMMAND: "fanotify07"
PID: 1459   TASK: ffff8800b8e5aa00  CPU: 2   COMMAND: "fanotify07"
PID: 1460   TASK: ffff88007f8cb800  CPU: 3   COMMAND: "fanotify07"
PID: 1461   TASK: ffff88007f8caa00  CPU: 2   COMMAND: "fanotify07"
PID: 1462   TASK: ffff88007f8ce200  CPU: 1   COMMAND: "fanotify07"
PID: 1463   TASK: ffff88007f8cf000  CPU: 2   COMMAND: "fanotify07"
PID: 1464   TASK: ffff88007f8c8e00  CPU: 1   COMMAND: "fanotify07"
PID: 1465   TASK: ffff88007f8cc600  CPU: 2   COMMAND: "fanotify07"
PID: 1466   TASK: ffff88007f509c00  CPU: 1   COMMAND: "fanotify07"
PID: 1467   TASK: ffff88007f50b800  CPU: 2   COMMAND: "fanotify07"
PID: 1468   TASK: ffff88007f50f000  CPU: 1   COMMAND: "fanotify07"
```

```
 #0 [ffff8800bb1cbcd8] __schedule at ffffffff8168a825
 #1 [ffff8800bb1cbd28] schedule at ffffffff8168adbb
 #2 [ffff8800bb1cbd38] fanotify_handle_event at ffffffff812396a3
 #3 [ffff8800bb1cbdb8] fsnotify at ffffffff81235ea9
 #4 [ffff8800bb1cbe88] security_file_permission at ffffffff812fc086
 #5 [ffff8800bb1cbeb8] rw_verify_area at ffffffff811f2ccf
 #6 [ffff8800bb1cbed8] vfs_read at ffffffff811f2dd6
 #7 [ffff8800bb1cbf10] sys_read at ffffffff811f3b85
 #8 [ffff8800bb1cbf50] entry_SYSCALL_64_fastpath at ffffffff8168eda5
```

Lets Cheat a bit, so you can focus in the Debugging skills, and not the problem

Debugging: Real Cases
BUG #3303 (fanotify07: source code to help you)

```c
static void test_fanotify(void)
{
        int newfd;
        int ret;

        fd_notify = setup_instance();
        run_children();
        loose_fanotify_events();

        /*
         * Create and destroy another instance. This may hang if
         * unanswered fanotify events block notification subsystem.
         */
        newfd = setup_instance();
        if (close(newfd)) {
                tst_brk(TBROK | TERRNO, "close(%d) failed", newfd);
        }

        tst_res(TPASS, "second instance destroyed successfully");

        /*
         * Now destroy the fanotify instance while there are permission
         * events at various stages of processing. This may provoke
         * kernel hangs or crashes.
         */
        SAFE_CLOSE(fd_notify);
        ret = stop_children();
        if (ret)
                tst_res(TFAIL, "child exited for unexpected reason");
        else
                tst_res(TPASS, "all children exited successfully");
}
```

PID: 1452

PID: 1451

```c
static int setup_instance(void)
{
        int fd;

        fd = SAFE_FANOTIFY_INIT(FAN_CLASS_CONTENT, O_RDONLY);

        if (fanotify_mark(fd, FAN_MARK_ADD, FAN_ACCESS_PERM, AT_FDCWD,
                          fname) < 0) {
                close(fd);
                if (errno == EINVAL) {
                        tst_brk(TCONF | TERRNO,
                                "CONFIG_FANOTIFY_ACCESS_PERMISSIONS not "
                                "configured in kernel?");
                } else {
                        tst_brk(TBROK | TERRNO,
                                "fanotify_mark (%d, FAN_MARK_ADD, FAN_ACCESS_PE
                                "AT_FDCWD, %s) failed.", fd, fname);
                }
        }

        return fd;
}
```

```c
static void run_children(void)
{
        int i;

        for (i = 0; i < MAX_CHILDREN; i++) {
                child_pid[i] = SAFE_FORK();
                if (!child_pid[i]) {
                        /* Child will generate events now */
                        close(fd_notify);
                        generate_events();
                        exit(0);
                }
        }
```

Debugging: Real Cases
BUG #3303 (fanotify07: source code to help you)

```c
static void loose_fanotify_events(void)
{
	int not_responded = 0;

	/*
	 * check events
	 */
	while (not_responded < MAX_NOT_RESPONDED) {
		struct fanotify_event_metadata event;
		struct fanotify_response resp;

		/* Get more events */
		SAFE_READ(1, fd_notify, &event, sizeof(event));

		if (event.mask != FAN_ACCESS_PERM) {
			tst_res(TFAIL,
				"get event: mask=%llx (expected %llx) "
				"pid=%u fd=%u",
				(unsigned long long)event.mask,
				(unsigned long long)FAN_ACCESS_PERM,
				(unsigned)event.pid, event.fd);
			break;
		}

		/*
		 * We respond to permission event with 95% percent
		 * probability. */
		if (random() % 100 > 5) {
			/* Write response to permission event */
			resp.fd = event.fd;
			resp.response = FAN_ALLOW;
			SAFE_WRITE(1, fd_notify, &resp, sizeof(resp));
		} else {
			not_responded++;
		}
```

```c
#define BUF_SIZE 256
static char fname[BUF_SIZE];
static char buf[BUF_SIZE];
static volatile int fd_notify;

/* Number of children we start */
#define MAX_CHILDREN 16
static pid_t child_pid[MAX_CHILDREN];

/* Number of children we don't respond to before stopping */
#define MAX_NOT_RESPONDED 4
```

```c
static void generate_events(void)
{
	int fd;

	/*
	 * generate sequence of events
	 */
	if ((fd = open(fname, O_RDWR | O_CREAT, 0700)) == -1)
		exit(1);

	/* Run until killed... */
	while (1) {
		lseek(fd, 0, SEEK_SET);
		if (read(fd, buf, BUF_SIZE) == -1)
			exit(3);
```

ALL OTHER TASKS

Feel better ? Let's continue, finding out things like we NEVER saw the source code

Debugging: Real Cases
BUG #3303 (test suite program stack traces)

```
PID: 1451   TASK: ffff8800b8c4e200  CPU: 2   COMMAND: "fanotify07"

 #0 [ffff8800bb2efe20] __schedule at ffffffff8168a825
 #1 [ffff8800bb2efe70] schedule at ffffffff8168adbb
 #2 [ffff8800bb2efe80] do_wait at ffffffff8108356f
 #3 [ffff8800bb2efec0] sys_wait4 at ffffffff810845e4
 #4 [ffff8800bb2eff50] entry_SYSCALL_64_fastpath at ffffffff8168eda5
```

- kernel entry point: wait()
- main task waiting for all children

```
PID: 1452   TASK: ffff8800b8c4aa00  CPU: 0   COMMAND: "fanotify07"

 #0 [ffff8800b8c47c48] __schedule at ffffffff8168a825
 #1 [ffff8800b8c47c98] schedule at ffffffff8168adbb
 #2 [ffff8800b8c47ca8] schedule_timeout at ffffffff8168de32
 #3 [ffff8800b8c47d30] wait_for_completion at ffffffff8168c299
 #4 [ffff8800b8c47d88] __synchronize_srcu at ffffffff810de59f
 #5 [ffff8800b8c47df8] synchronize_srcu at ffffffff810de5f4
 #6 [ffff8800b8c47e10] fsnotify_destroy_group at ffffffff8123681b
 #7 [ffff8800b8c47e30] fanotify_release at ffffffff812399b0
 #8 [ffff8800b8c47e78] __fput at ffffffff811f4907
 #9 [ffff8800b8c47ec0] ____fput at ffffffff811f4aae
#10 [ffff8800b8c47ed0] task_work_run at ffffffff8109db7f
#11 [ffff8800b8c47f08] exit_to_usermode_loop at ffffffff81003233
#12 [ffff8800b8c47f30] syscall_return_slowpath at ffffffff81003c6e
#13 [ffff8800b8c47f50] int_ret_from_sys_call at ffffffff8168ef2b
```

- no user stack trace, but…
- we know its returning from kernel to userland!

```
PID: 1453   TASK: ffff8800b8e58e00  CPU: 2   COMMAND: "fanotify07"
...
PID: 1468   TASK: ffff88007f50f000  CPU: 1   COMMAND: "fanotify07"

 #0 [ffff8800bb1cbcd8] __schedule at ffffffff8168a825
 #1 [ffff8800bb1cbd28] schedule at ffffffff8168adbb
 #2 [ffff8800bb1cbd38] fanotify_handle_event at ffffffff812396a3
 #3 [ffff8800bb1cbdb8] fsnotify at ffffffff81235ea9
 #4 [ffff8800bb1cbe88] security_file_permission at ffffffff812fc086
 #5 [ffff8800bb1cbeb8] rw_verify_area at ffffffff811f2ccf
 #6 [ffff8800bb1cbed8] vfs_read at ffffffff811f2dd6
 #7 [ffff8800bb1cbf10] sys_read at ffffffff811f3b85
 #8 [ffff8800bb1cbf50] entry_SYSCALL_64_fastpath at ffffffff8168eda5
```

- kernel entry point: read()
- vfs_read() ⇨ rw_verify_area()
- we know where to focus!

SUMMARY

1. main task waiting for its children
2. one task blocked when ret from kernel to userland
3. all other tasks blocked while trying to read() a file

Debugging: Real Cases
BUG #3303 (fanotify07: 1 trace calls attention)

```
PID: 1452   TASK: ffff8800b8c4aa00  CPU: 0   COMMAND: "fanotify07"

 #1 [ffff8800b8c47c98] schedule at ffffffff8168adbb
 #2 [ffff8800b8c47ca8] schedule_timeout at ffffffff8168de32
 #3 [ffff8800b8c47d30] wait_for_completion at ffffffff8168c299

      - WAIT: SRCU read-side critical-section completion              "BLOCKS"

 #4 [ffff8800b8c47d88] __synchronize_srcu at ffffffff810de59f
 #5 [ffff8800b8c47df8] synchronize_srcu at ffffffff810de5f4

      - WAIT: prior srcu read-side critical-section to complete
      - WAIT: the count of both indexes to drain to zero (internal SRCU structures)
      - NOTE: ILLEGAL to call it from corresponding srcu read-side critical

 #6 [ffff8800b8c47e10] fsnotify_destroy_group at ffffffff8123681b

      fsnotify_group_stop_queueing(group);
      fsnotify_clear_marks_by_group(group);
      synchronize_srcu(&fsnotify_mark_srcu);                          "BLOCKS"
                       |
                       |   RCU STRUCT initialized by module initialization (fsnotify_init)
                       |   WHO USES IT:
                       |
                  fsnotify_init            - creates the sleep rcu
                  fsnotify                 - main function called by all vfs fops hooks, read-side critical
                  fsnotify_mark_destroy    - BLOCKED (active) in KERNEL THREAD
                  fsnotify_destroy_group   - BLOCKED (active) in fanotify07
```

THIS SLIDE

Debugging: Real Cases
BUG #3303 (Inside crash: Userland tasks)

```
#7 [ffff8800b8c47e30] fanotify_release at ffffffff812399b0
```

```
group comes from fd->private_data
destroy group is the last function before release returns
```

allows all remaining permission events (access list and queue) and simulate reply from userspace

```
locks the group->fanotify_data
        for each existing event in fanotify_data.access_list:
                removes event from event->fae.fse.list
                responds event with FAN_ALLOW
unlocks group->fanotify_data
```

gets events from notify queue and: if no permission, destroy, if permission allows (set response)

```
mutex_lock(&group->notification_mutex);
while (!fsnotify_notify_queue_is_empty(group)) {
        fsn_event = fsnotify_remove_first_event(group);
        if (!(fsn_event->mask & FAN_ALL_PERM_EVENTS))
                fsnotify_destroy_event(group, fsn_event);
        else
                FANOTIFY_PE(fsn_event)->response = FAN_ALLOW;
}
mutex_unlock(&group->notification_mutex);
```

wakes wait queue

```
wake_up(&group->fanotify_data.access_waitq);

/* matches the fanotify_init->fsnotify_alloc_group */

fanotify_destroy_group(group);
```

"BLOCKS"

WAIT! Let's understand RCU and SRCU before moving further

Feel better ? Let's **continue**, like if we never **stopped** =)

Debugging: Real Cases
BUG #3303 (look for fsnotify_mark_destroy)

```
crash> foreach bt
...
PID: 7      TASK: ffff88013ab1d400  CPU: 1   COMMAND: "rcu_sched"
 #0 [ffff88013abb3d58] __schedule at ffffffff8168a825
 #1 [ffff88013abb3da8] schedule at ffffffff8168adbb
 #2 [ffff88013abb3db8] schedule_timeout at ffffffff8168dd0c
 #3 [ffff88013abb3e40] rcu_gp_kthread at ffffffff810e1949
 #4 [ffff88013abb3ec0] kthread at ffffffff8109f617
 #5 [ffff88013abb3f50] ret from fork at ffffffff8168f1a2
PID: 8      TASK: ffff88013ab1e200  CPU: 0   COMMAND: "rcu_bh"
 #0 [ffff88013abb7de0] __schedule at ffffffff8168a825
 #1 [ffff88013abb7e30] schedule at ffffffff8168adbb
 #2 [ffff88013abb7e40] rcu_gp_kthread at ffffffff810e1e0c
 #3 [ffff88013abb7ec0] kthread at ffffffff8109f617
 #4 [ffff88013abb7f50] ret from fork at ffffffff8168f1a2
PID: 41     TASK: ffff88013a55e200  CPU: 0   COMMAND: "fsnotify_mark"
 #0 [ffff8800b8e1fc78] __schedule at ffffffff8168a825
 #1 [ffff8800b8e1fcc8] schedule at ffffffff8168adbb
 #2 [ffff8800b8e1fcd8] schedule_timeout at ffffffff8168de32
 #3 [ffff8800b8e1fd60] wait_for_completion at ffffffff8168c299
 #4 [ffff8800b8e1fdb8] __synchronize_srcu at ffffffff810de59f
 #5 [ffff8800b8e1fe28] synchronize_srcu at ffffffff810de5f4
 #6 [ffff8800b8e1fe40] fsnotify_mark_destroy at ffffffff81236e05
 #7 [ffff8800b8e1fec0] kthread at ffffffff8109f617
 #8 [ffff8800b8e1ff50] ret_from_fork at ffffffff8168f1a2
```

KERNEL THREAD

KERNEL THREAD

KERNEL THREAD

"BLOCKS"

Debugging: Real Cases
BUG #3303 (kthread with fsnotify_mark_srcu)

```
PID: 41     TASK: ffff88013a55e200  CPU: 0   COMMAND: "fsnotify_mark"   KERNEL THREAD
 #0 [xx] __schedule at ffffffff8168a825
 #1 [xx] schedule at ffffffff8168adbb
 #2 [xx] schedule_timeout at ffffffff8168de32
 #3 [xx] wait_for_completion at ffffffff8168c299
 #4 [xx] __synchronize_srcu at ffffffff810de59f
 #5 [xx] synchronize_srcu at ffffffff810de5f4
 #6 [xx] fsnotify_mark_destroy at ffffffff81236e05
 #7 [xx] kthread at ffffffff8109f617
 #8 [xx] ret_from_fork at ffffffff8168f1a2
```

- completion is always finished somewhere else
- waiting for srcu read-side critical-section compl.
- **"BLOCKS"**

SRCU is waiting on its **OWN completion** (from __synchronize_srcu())

- completion = rcu_batch_queue **batch_check0**
- schedules a **wakeme_after_rcu()** to FINALLY return, after completion, wait **batch_check0**
- since it scheduled a **rcu_head** at the end of the queue, it will GUARANTEE **all work was done**!

- **EXPECTED: synchronize_srcu() finally returns**
- synchronize_srcu() calls **srcu_advance_batches()**:
  - it **moves callbacks** from:
    - batch_check0 to
    - batch_check1 and
    - batch_done as readers drain

Debugging: Real Cases
BUG #3303 (Inside crash: Kernel explained)

2 TASKS WAITING ON RCU CALLBACKS

Regular task **fanotify07** is STUCK

- it is stuck at **fsnotify_destroy_group()**, which tried to sync RCU
- called **synchronize_rcu()** and it is waiting RCU (internal) completion
- being **1st** who **called RCU sync**, it would be on **its own grace period**
- being **2nd who called RCU sync**, it would have been **scheduled** for **next grace period**

Kernel thread **fsnotify_mark** is STUCK

- it is stuck at **fsnotify_mark_destroy()**, which tried to sync RCU
- called **synchronize_rcu()** and it is waiting its completion
- being **1st** who **called RCU sync**, it would be on **its own grace period**
- being **2nd who called RCU sync**, it would have been **scheduled** for **next grace period**

Differential Diagnosis: TO PLAN

Pretend you are Dr House =)

Debugging: Real Cases
BUG #3303 (Differential Diagnosis: TO PLAN)

What is DEFINITELY a certainty ?

- SRCU is "**blocked**" waiting for the **completion** of ALL **scheduled** callbacks of ONE **graceful period**.

Where to go ?

1

To discover if SRCU is the problem:

- Find out size of current's grace period queue, callbacks scheduled in each CPU cb queue
- Find out size of next grace period queue (callbacks being enqueued for a 2nd synchronize_srcu())
- Find out if there is any ongoing interrupt (IPI) dealing with CPU synchronization (lost completion ?)

OBS: We could, for example, find a race with grace period enqueue logic

2

To discover if SRCU callers are the problem:

- Check SRCU internal structures for how many read locks are held
- Compare with the number of locked tasks and check if it is the same

My mission:

- Find a way to unblock this logic (even if not backed by recent (mainline/next) upstream change

Debugging: Real Cases
BUG #3303 (Why to focus in SRCU internals ?)

- I got history with RCU stalls on vCPUs (+ IPI issues in the past):

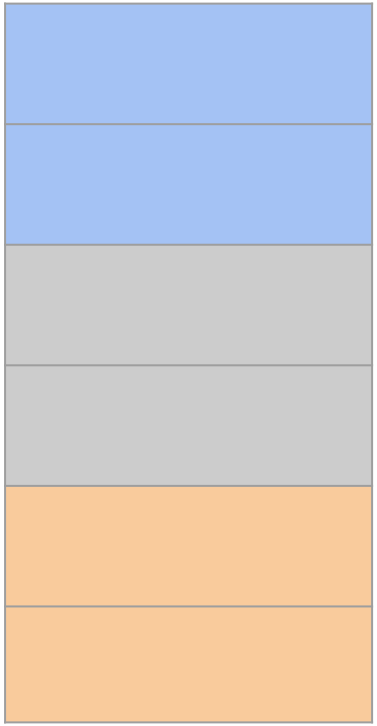
  https://github.com/rafaeldtinoco/work/blob/master/notes/handmade/rcu-contention-for-vcpus-with-steal-time.pdf

- Diagnosis is ALWAYS based on your personal past experiences!

- That explains why, unfortunately for this case, I haven't done obvious: Having 2 SRCU synchronization calls, one directly from the fanotify07 task, and other from one kernel thread doing fanotify07 deferred work seemed suspicious.

	CPU 0	CPU 1	CPU 2	CPU 3
1	`i0 = srcu_read_lock(&s1);`			`i3 = srcu_read_lock(&s2);`
2	read-side critical session	`synchronize_srcu(&s1);` `[ ENTER ]`		
3		wait time (sleeping)	`i2 = srcu_read_lock(&s1);`	
4	`srcu_read_unlock(&s1, i0);`		read-side critical session	
5		`synchronize_srcu(&s1);` `[ EXIT ]`		
6			`srcu_read_unlock(&s1, i2);`	

SRCU **updates** and **read-side critical sections**



SRCU **updates** and **read-side critical sections**

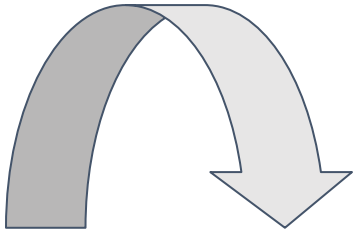
```
static void __synchronize_srcu(struct srcu_struct *sp, int trycount)
{
        struct rcu_synchronize rcu;
        struct rcu_head *head = &rcu.head;
…

        might_sleep();
        init_completion(&rcu.completion);

        head->next = NULL;
        head->func = wakeme_after_rcu;
        spin_lock_irq(&sp->queue_lock);

        if (!sp->running) {
                sp->running = true;

                rcu_batch_queue(&sp->batch_check0, head);
                spin_unlock_irq(&sp->queue_lock);

                srcu_advance_batches(sp, trycount);

                if (!rcu_batch_empty(&sp->batch_done)) {
                        BUG_ON(sp->batch_done.head != head);
                        rcu_batch_dequeue(&sp->batch_done);
                        done = true;
                }

                srcu_reschedule(sp);
        } else {
                rcu_batch_queue(&sp->batch_queue, head);
                spin_unlock_irq(&sp->queue_lock);
        }

        if (!done)
                wait_for_completion(&rcu.completion);
}
```

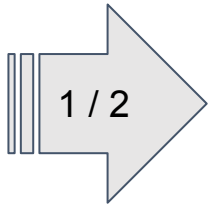
PROBLEM

WAIT FOR COMPLETION NEVER RETURNED:

1. SRCU BATCH NEVER FINISHED
2. LAST CALLBACK WOULD HAVE UNBLOCKED IT

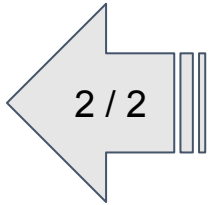
THIS WILL ASYNC UNBLOCK THIS FUNCTION AS THE LAST SCHEDULED BATCH





```
static void srcu_reschedule(struct srcu_struct *sp)
{
      bool pending = true;

      if (rcu_batch_empty(&sp->batch_done) &&
          rcu_batch_empty(&sp->batch_check1) &&
          rcu_batch_empty(&sp->batch_check0) &&
          rcu_batch_empty(&sp->batch_queue)) {

            spin_lock_irq(&sp->queue_lock);

            if (rcu_batch_empty(&sp->batch_done) &&
                rcu_batch_empty(&sp->batch_check1) &&
                rcu_batch_empty(&sp->batch_check0) &&
                rcu_batch_empty(&sp->batch_queue)) {

                  sp->running = false;
                  pending = false;
            }

            spin_unlock_irq(&sp->queue_lock);
      }

      if (pending)
            queue_delayed_work(system_power_efficient_wq,
                        &sp->work, SRCU_INTERVAL);
}
```

```
void wakeme_after_rcu(struct rcu_head *head)
{
      struct rcu_synchronize *rcu;

      rcu = container_of(head, struct rcu_synchronize,
                         head);
      complete(&rcu->completion);
}
```



Debugging: Real Cases
BUG #3303 (SRCU problem ?)

Functions that will **refer this WORK QUEUE**:

- `call_srcu()`
  callees are unknown to any of our stack traces
- `init_srcu_struct_fields()`
  initializes SRCU_STRUCT WORKQUEUE
- `srcu_reschedule()`
  schedules this WORKQUEUE, called by:
  `__synchronize_srcu()`
  `synchronize_srcu()`
  `fs_notify_destroy_group()`
  `fs_notify_mark_destroy()`

Functions dealing with same SRCU section ?

YES -> fsnotify_mark_srcu

So, who is using this specific **SRCU** then ?

- `fsnotify_init()`
  `init_srcu_struct_fields()`
- `fsnotify_destroy_group()`
  `synchronize_srcu()`
- `fsnotify_mark_destroy()`
  `synchronize_srcu()`

1

- `fsnotify()`

2

```
srcu_read_lock(s1)
...
srcu_dereference(x,&s1)
...
srcu_read_unlock(s1)
```

CRITICAL READ-SIDE SECTION

Debugging: Real Cases
BUG #3303 (SRCU problem ?)


```
static __init int fsnotify_init(void)
{
…
        ret = init_srcu_struct(&fsnotify_mark_srcu);
        if (ret)
                panic("initializing fsnotify_mark_srcu");
…
}
```

and the initialization of "fsnotify_mark_srcu" is actually:

```
static int init_srcu_struct_fields(struct srcu_struct *sp)
{
        sp->completed = 0;
        spin_lock_init(&sp->queue_lock);
        sp->running = false;
        rcu_batch_init(&sp->batch_queue);
        rcu_batch_init(&sp->batch_check0);
        rcu_batch_init(&sp->batch_check1);
        rcu_batch_init(&sp->batch_done);
        INIT_DELAYED_WORK(&sp->work, process_srcu); ---------------- WORK QUEUE HANDLING RCU PERIODS
        sp->per_cpu_ref = alloc_percpu(struct srcu_struct_array);
        return sp->per_cpu_ref ? 0 : -ENOMEM;
}
```

Functions that will **refer this WORK QUEUE**:

- `call_srcu()`
  callees are unknown to any of our stack traces
- `init_srcu_struct_fields()`
  initializes SRCU_STRUCT WORKQUEUE
- `srcu_reschedule()`
  schedules this WORKQUEUE, called by:
  `__synchronize_srcu()`
  `synchronize_srcu()`
  `fs_notify_destroy_group()`
  `fs_notify_mark_destroy()`

Debugging: Real Cases
BUG #3303 (SRCU problem ?)

DISCONSIDER THIS SLIDE FOR THE CASE. JUST SHOWING HOW CALLBACKS WORK
call_srcu() is not called in this specific scenario

call_srcu(); Enqueues a SRCU cb on specific srcu structure, initiating a grace period!!

```
void call_srcu(struct srcu_struct *sp, struct rcu_head *head, rcu_callback_t func)
{
     unsigned long flags;

     head->next = NULL;
     head->func = func;
     spin_lock_irqsave(&sp->queue_lock, flags);
     rcu_batch_queue(&sp->batch_queue, head);
     if (!sp->running) {
          sp->running = true;
          queue_delayed_work(system_power_efficient_wq, &sp->work, 0);
     }
     spin_unlock_irqrestore(&sp->queue_lock, flags);
}
```

Debugging: Real Cases
BUG #3303 (SRCU problem ?)

```
static int __init fsnotify_mark_init(void)
{
	struct task_struct *thread;

	thread = kthread_run(fsnotify_mark_destroy, NULL, "fsnotify_mark");

	if (IS_ERR(thread))
		panic("unable to start fsnotify mark destruction thread.");

	return 0;
}
```

Debugging: Real Cases
BUG #3303 (SRCU problem ?)

```
static int fsnotify_mark_destroy(void *ignored)
{
      struct fsnotify_mark *mark, *next;
      struct list_head private_destroy_list;

      for (;;) {
            spin_lock(&destroy_lock);
            list_replace_init(&destroy_list, &private_destroy_list); ---- replaces OLD → NEW list
            spin_unlock(&destroy_lock);                                    re-init OLD list

            synchronize_srcu(&fsnotify_mark_srcu);       ---------------------- WAIT READ-SIDE
SECTIONS

            list_for_each_entry_safe(mark, next, &private_destroy_list, g_list) {
                  list_del_init(&mark->g_list);
                  fsnotify_put_mark(mark);
            }

            wait_event_interruptible(destroy_waitq, !list_empty(&destroy_list));
      }

      return 0;
```

Debugging: Real Cases
BUG #3303 (SRCU problem ?)

```
void fsnotify_destroy_group(struct fsnotify_group *group)
{
      /* Stop queueing new events. The code below is careful enough to not require this but fanotify
         needs to stop queuing events even before fsnotify_destroy_group() is called and this makes
         the other callers of fsnotify_destroy_group() to see the same behavior. */

       fsnotify_group_stop_queueing(group);

       /* clear all inode marks for this group */
       fsnotify_clear_marks_by_group(group);

       synchronize_srcu(&fsnotify_mark_srcu);

       /* clear the notification queue of all events */
       fsnotify_flush_notify(group);

       /* Destroy overflow event (we cannot use fsnotify_destroy_event() as that deliberately
          ignores overflow events. */

       if (group->overflow_event)
             group->ops->free_event(group->overflow_event);

       fsnotify_put_group(group);
}
```

Debugging: Real Cases
BUG #3303 (SRCU problem ?)

```
INIT_DELAYED_WORK(&sp->work, process_srcu); ---------------- WORK QUEUE HANDLING RCU PERIODS
```

INIT_DELAYED_WORK TRANSLATES INTO:

```
do {
        do {
                __init_work(((&(&sp->work)->work)), 0);
                ((&(&sp->work)->work))->data = (atomic_long_t) { (WORK_STRUCT_NO_POOL) };
                INIT_LIST_HEAD(&((&(&sp->work)->work))->entry);
                ((&(&sp->work)->work))->func = ((process_srcu))); ----------- WORK QUEUE FUNCTION
        } while (0);

        do {
                init_timer_key(((&(&sp->work)->timer)), ...);
                (&(&sp->work)->timer)->function = (delayed_work_timer_fn); --- FUNCTION TO EXEC AFTER TIMEOUT
                (&(&sp->work)->timer)->data = ((unsigned long)(&sp->work)); -- ARGUMENT TO GIVE TO FUNCTION

        } while (0);

} while (0)
```

Debugging: Real Cases
BUG #3303 (SRCU problem ?)

```
(&(&sp->work)->timer)->function = (delayed_work_timer_fn); ---
(&(&sp->work)->timer)->data = ((unsigned long)(&sp->work)); --
((&(&sp->work)->work))->func = (((process_srcu))); ----------
```

FUNCTION TO EXEC AFTER TIMEOUT
ARGUMENT TO GIVE TO FUNCTION
WORK QUEUE FUNCTION

This is the WORK QUEUE FUNCTION that handles SRCU grace periods:

```
void process_srcu(struct work_struct *work)
{
      struct srcu_struct *sp;

      sp = container_of(work, struct srcu_struct, work.work);

      srcu_collect_new(sp);
            Move any new SRCU callbacks to the first stage of the SRCU grace period pipeline.
      srcu_advance_batches(sp, 1);
            Core SRCU state machine. cbs from ->batch_check0 to ->batch_check then ->batch_done.
      srcu_invoke_callbacks(sp);
            Runs limited number of callbacks that have passed their grace period. Can reschedule.
      srcu_reschedule(sp);
            Finished one SRCU round. Start another if cbs queued, if not, put SRCU no-running.
}
```

Debugging: Real Cases
BUG #3303 (SRCU problem ?)


```
srcu_collect_new(sp); -------- Move any new SRCU callbacks to the first stage of the SRCU grace period pipeline.

static void srcu_collect_new(struct srcu_struct *sp)
{
	if (!rcu_batch_empty(&sp->batch_queue)) {
		spin_lock_irq(&sp->queue_lock);
		rcu_batch_move(&sp->batch_check0, &sp->batch_queue);
		spin_unlock_irq(&sp->queue_lock);
	}
}

srcu_invoke_callbacks(sp); --- Runs limited number of callbacks that have passed their grace period. Can reschedule.

static void srcu_invoke_callbacks(struct srcu_struct *sp)
{
	int i;
	struct rcu_head *head;

	for (i = 0; i < SRCU_CALLBACK_BATCH; i++) {
		head = rcu_batch_dequeue(&sp->batch_done);
		if (!head)
			break;
		local_bh_disable();
		head->func(head);
		local_bh_enable();
	}
```

Debugging: Real Cases
BUG #3303 (SRCU problem ?)

- Find out if there is any ongoing interrupt (IPI) dealing with CPU synchronization
- Find out size of current's grace period queue, callbacks scheduled in each CPU cb queue
- Find out size of next grace period queue
  (callbacks being enqueued for a 2nd synchronize_srcu()).

fsnotify_mark_srcu - timers and callbacks from kdump

SUMMARY: Basically showing graceful period is on-going AND the that the 2 callbacks that are going to finish the completions being held are scheduled for the next 2 SRCU graceful periods.

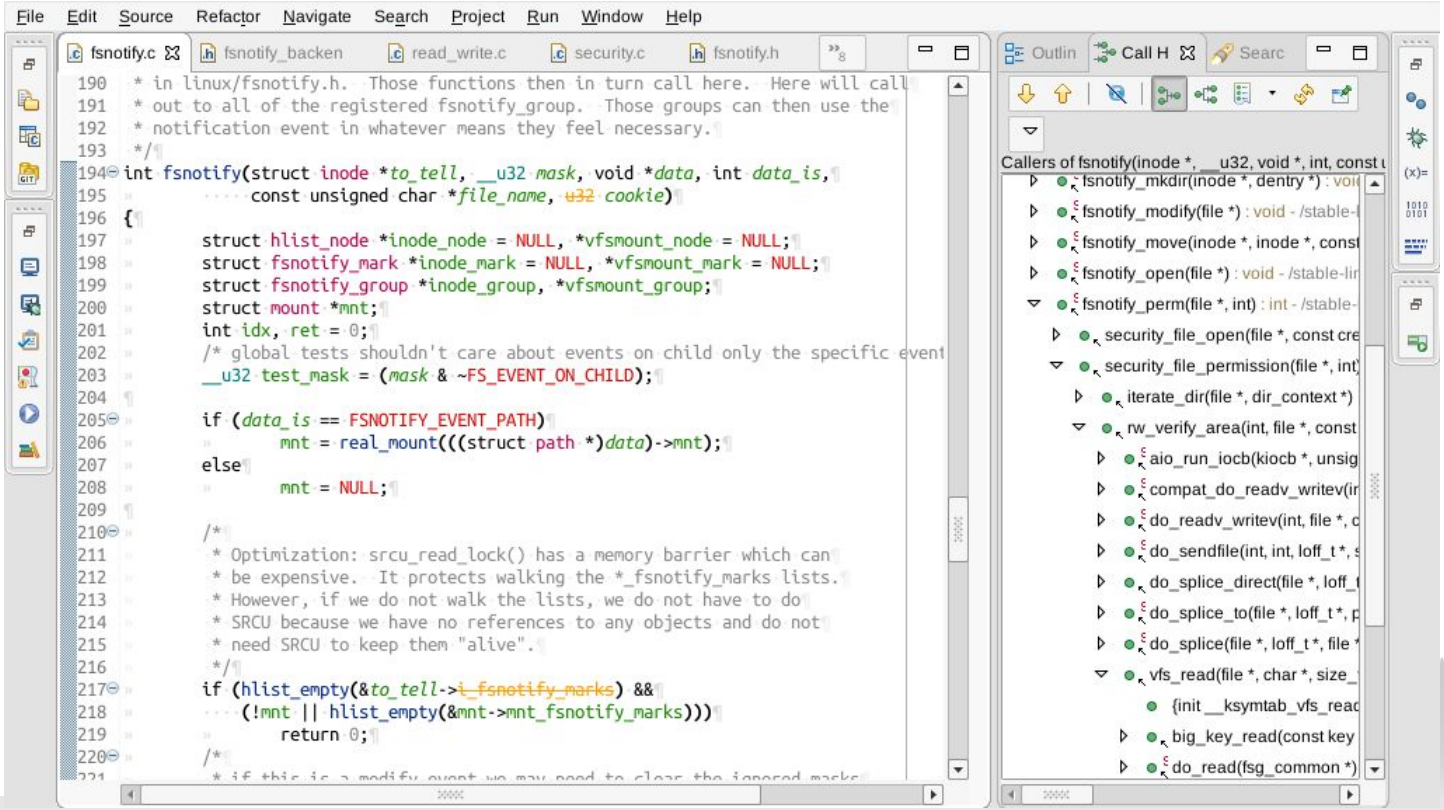
It is likely a CALLER issue then...

Debugging: Real Cases
BUG #3303 (SRCU caller problem ?)

- Check SRCU internal structures for how many read locks are held

```
PID: 1453   TASK: ffff8800b8e58e00  CPU: 2   COMMAND: "fanotify07"
...
PID: 1468   TASK: ffff88007f50f000  CPU: 1   COMMAND: "fanotify07"

 #0 [ffff8800bb1cbcd8] __schedule at ffffffff8168a825
 #1 [ffff8800bb1cbd28] schedule at ffffffff8168adbb
 #2 [ffff8800bb1cbd38] fanotify_handle_event at ffffffff812396a3
 #3 [ffff8800bb1cbdb8] fsnotify at ffffffff81235ea9
 #4 [ffff8800bb1cbe88] security_file_permission at
ffffffff812fc086
 #5 [ffff8800bb1cbeb8] rw_verify_area at ffffffff811f2ccf
 #6 [ffff8800bb1cbed8] vfs_read at ffffffff811f2dd6
 #7 [ffff8800bb1cbf10] sys_read at ffffffff811f3b85
 #8 [ffff8800bb1cbf50] entry_SYSCALL_64_fastpath at
ffffffff8168eda5
```

Debugging: Real Cases
BUG #3303 (SRCU caller problem ?)



Debugging: Real Cases
BUG #3303 (SRCU caller problem ?)

- Check SRCU internal structures for how many read locks are held

  fanotify_handle_event - SRCU callers and summary
  fanotify_handle_event - SRCU read locks being held

  SUMMARY: Second link concludes that the number of read locks being held is compatible with the number of tasks in uninterruptible state from fanotify07 LTP test.

  It shows that there is a RACE in the fsnotify logic: if userland stops responding the security events and fanotify is asked to

Debugging: Real Cases
BUG #3303 (Final Result)

- Summary (fanotify07 failing for v4.9 and v4.4 in LKFT tests)
  https://bugs.linaro.org/show_bug.cgi?id=3303#c15

- Led me to
  https://www.spinics.net/lists/linux-fsdevel/msg109131.html

- Which made me prepare the backport (next slide)
  rafaeldtinoco/work/sources/patches/bugs/3303

- And proposed it upstream
  https://www.spinics.net/lists/stable/msg247115.html

The End: Upstream decided this patchset was too intrusive to be backported to v4.9, so we, Linaro KVT, kept all bugs documented and patchset ready, if ever needed by anyone.
We marked fanotify07 test as skipped for v4.4 and v4.9 in our LKFT.

kernel v4.9 backport

```
** [PATCH 35/35] 054c636e5c8054884ede889be82ce059879945e6 fsnotify: Move ->free_mark callback to fsnotify_ops
ok [PATCH 34/35] 7b1293234084ddb6469c4e9a5ef818f399b5786b fsnotify: Add group pointer in fsnotify_init_mark()
ok [PATCH 33/35] ebb3b47e37a4ccef33e6388589a21a5c23d6b40b fsnotify: Drop inode_mark.c
ok [PATCH 32/35] b1362edfe15b20edd3d116cec521aa420b7afb98 fsnotify: Remove fsnotify_find_{inode|vfsmount}_mark()
ok [PATCH 31/35] 2e37c6ca8d76c362e844c0cf3ebe8ba2e27940cb fsnotify: Remove fsnotify_detach_group_marks()
ok [PATCH 30/35] 18f2e0d3a43641889ac2ba9d7508d47359eec063 fsnotify: Rename fsnotify_clear_marks_by_group_flags()
ok [PATCH 29/35] 416bcdbcbbb4800f11f03e8baf570f9996219f67 fsnotify: Inline fsnotify_clear_{inode|vfsmount}_mark_group()
ok [PATCH 28/35] 8920d2734d9a1b68e1b53d8c12b289773cdbd971 fsnotify: Remove fsnotify_recalc_{inode|vfsmount}_mask()
ok [PATCH 27/35] 66d2b81bcb92c14b22a56a9ff936f2b40accc83c fsnotify: Remove fsnotify_set_mark_{,ignored_}mask_locked()
ok [PATCH 26/35] 05f0e38724e8449184acd8fbf0473ee5a07adc6c fanotify: Release SRCU lock when waiting for userspace response
ok [PATCH 25/35] 9385a84d7e1f658bb2d96ab798393e4b16268aaa fsnotify: Pass fsnotify_iter_info into handle_event handler
** [NEEDED     ] 3cd5eca8d7a2fe43098df4c33a1272fe6945cac9 fsnotify: constify 'data' passed to ->handle_event()
ok [PATCH 24/35] abc77577a669f424c5d0c185b9994f2621c52aa4 fsnotify: Provide framework for dropping SRCU lock in ->handle_event
ok [PATCH 23/35] f09b04a03e0239f65bd964a1de758e53cf6349e8 fsnotify: Remove special handling of mark destruction on group shutdown
ok [PATCH 22/35] 6b3f05d24d355f50f3d9814304650fcab0efb482 fsnotify: Detach mark from object list when last reference is dropped
ok [PATCH 21/35] 11375145a70d69e871dd5b8fcadd5d1ee4162e7c fsnotify: Move queueing of mark for destruction into fsnotify_put_mark()
ok [PATCH 20/35] e7253760587e8523fe1e8ede092a620f1403f2e8 inotify: Do not drop mark reference under idr_lock
ok [PATCH 19/35] 08991e83b7286635167bab40927665a90fb00d81 fsnotify: Free fsnotify_mark_connector when there is no mark attached
ok [PATCH 18/35] 04662cab59fc3e8421fd7a0539d304d51d2750a4 fsnotify: Lock object list with connector lock
ok [PATCH 17/35] 2629718dd26f89e064dcdec6c8e5b9713502e1f8 fsnotify: Remove useless list deletion and comment
ok [PATCH 16/35] 73cd3c33ab793325ebaae27fa58b4f713c16f12c fsnotify: Avoid double locking in fsnotify_detach_from_object()
ok [PATCH 15/35] 8212a6097a720896b4cdbe516487ad47f4296599 fsnotify: Remove indirection from fsnotify_detach_mark()
ok [PATCH 14/35] a03e2e4f078365428bb4317989cb5d1d6563cfe9 fsnotify: Determine lock in fsnotify_destroy_marks()
ok [PATCH 13/35] f06fd98759451876f51607f60abd74c89b141610 fsnotify: Move locking into fsnotify_find_mark()
ok [PATCH 12/35] a242677bb1e6faa9bd82bd33afb2621071258231 fsnotify: Move locking into fsnotify_recalc_mask()
ok [PATCH 11/35] 0810b4f9f207910d90aee56d312d25f334796363 fsnotify: Move fsnotify_destroy_marks()
ok [PATCH 10/35] 755b5bc681eb46de7bfaec196f85e30efd95bd9f fsnotify: Remove indirection from mark list addition
ok [PATCH 09/35] e911d8af87dba7642138f4320ca3db80629989f2 fsnotify: Make fsnotify_mark_connector hold inode reference
ok [PATCH 08/35] 86ffe245c430f07f95d5d28d3b694ea72f4492e7 fsnotify: Move object pointer to fsnotify_mark_connector
ok [NEEDED     ] be29d20f3f5db1f0b4e49a4f6eeedf840e2bf9b1 audit: Fix sleep in atomic
ok [NEEDED     ] e3ba730702af370563f66cb610b71aa0ca67955e fsnotify: Remove fsnotify_duplicate_mark()
** [PATCH 07/35] 9dd813c15b2c101168808d4f5941a29985758973 fsnotify: Move mark list head from object into dedicated structure -> THIS ONE
ok [PATCH 06/35] c1f33073ac1b33510e956de7181438515e438db0 fsnotify: Update comments
ok [PATCH 05/35] 43471d15df0e7c40ca4df1513fc1dcf5765396ac audit_tree: Use mark flags to check whether mark is alive
ok [PATCH 04/35] f410ff65548c548fed5f7e38c4ef57a73ebfe3bd audit: Abstract hash key handling
ok [PATCH 03/35] c97476400d3b73376fc055e828d7388d6b9ea99a fanotify: Move recalculation of inode / vfsmount mask under mark_mutex
ok [PATCH 02/35] 25c829afbd74fb9594d2351d9e41be05bacb9903 inotify: Remove inode pointers from debug messages
ok [PATCH 01/35] 5198adf649a0b7b0f9ddb98b264e57b41516116b fsnotify: Remove unnecessary tests when showing fdinfo

ok = cherry-pick
** = backport
```

The End

Perguntas ? Obrigado!

Rafael David Tinoco - rafael.tinoco@linaro.org
Kernel Validation Team - Linaro

Oooookay… moving on...

Debugging: Real Cases
BUG #3903

LTP fs test suite running read_all test causing kernel crash on Hikey devices. This bug is pretty much reproducible on Hikey running mainline kernel.

kernel bug log:

```
----------------
[ 1274.758398] Internal error: synchronous external abort: 96000210
[ 1274.797360] CPU: 2 PID: 7883 Comm: read_all Not tainted 4.17.0 #1
[ 1274.805577] Hardware name: HiKey Development Board (DT)
[ 1274.812990] pstate: 20000085 (nzCv daIf -PAN -UAO)
[ 1274.820029] pc : regmap_mmio_read32le+0x24/0x38
[ 1274.826856] lr : regmap_mmio_read+0x48/0x70
...
[ 1274.957914] Process read_all (pid: 7883, …
[ 1274.967757] Call trace:
[ 1274.973178]  regmap_mmio_read32le+0x24/0x38
[ 1274.980416]  regmap_mmio_read+0x48/0x70
[ 1274.987322]  _regmap_bus_reg_read+0x38/0x48
[ 1274.994625]  _regmap_read+0x74/0x2b8
[ 1275.001332]  regmap_read+0x50/0x78
[ 1275.007907]  regmap_read_debugfs+0x198/0x2e8
[ 1275.015396]  regmap_map_read_file+0x48/0x58
[ 1275.022834]  full_proxy_read+0x68/0x98
[ 1275.029858]  __vfs_read+0x60/0x170
[ 1275.036558]  vfs_read+0x94/0x150
[ 1275.043116]  ksys_read+0x6c/0xd8
[ 1275.049693]  sys_read+0x34/0x48
[ 1275.056217]  __sys_trace_return+0x0/0x4
[ 1275.063490] Code: aa1e03e0 d503201f f9400280 8b334000 (b9400000)
[ 1275.073144] ---[ end trace dd4f5045a6fc847a ]---
```

```
[ 1275.081364] BUG: sleeping function called from invalid context at
/srv/oe/build/tmp-rpb-glibc/work-shared/hikey/kernel-source/include/li
nux/percpu-rwsem.h:34
[ 1275.099461] in_atomic(): 1, irqs_disabled(): 128, pid: 7883, name:
read_all
...
[ 1275.179618] CPU: 2 PID: 7883 Comm: read_all Tainted: G      D
4.17.0 #1
[ 1275.191744] Hardware name: HiKey Development Board (DT)
[ 1275.201647] Call trace:
[ 1275.208780]  dump_backtrace+0x0/0x170
[ 1275.217188]  show_stack+0x24/0x30
[ 1275.225271]  dump_stack+0xac/0xe4
[ 1275.233370]  ___might_sleep+0x1c0/0x1f0
[ 1275.242043]  __might_sleep+0x58/0x90
[ 1275.250489]  exit_signals+0x4c/0x2d0
[ 1275.258954]  do_exit+0xb0/0xb40
[ 1275.267005]  die+0x1d4/0x200
[ 1275.274830]  arm64_notify_die+0x88/0xa0
[ 1275.283676]  do_sea+0xe0/0x150
[ 1275.291753]  do_mem_abort+0x68/0x110
[ 1275.300402]  el1_da+0x20/0x80
[ 1275.308464]  regmap_mmio_read32le+0x24/0x38
[ 1275.317805]  regmap_mmio_read+0x48/0x70
[ 1275.326817]  _regmap_bus_reg_read+0x38/0x48
[ 1275.336225]  _regmap_read+0x74/0x2b8
[ 1275.345043]  regmap_read+0x50/0x78
[ 1275.353735]  regmap_read_debugfs+0x198/0x2e8
[ 1275.363325]  regmap_map_read_file+0x48/0x58
[ 1275.372863]  full_proxy_read+0x68/0x98
[ 1275.381890]  __vfs_read+0x60/0x170
[ 1275.390248]  vfs_read+0x94/0x150
[ 1275.398091]  ksys_read+0x6c/0xd8
[ 1275.405871]  sys_read+0x34/0x48
[ 1275.413531]  __sys_trace_return+0x0/0x4
[ 1275.421966] note: read_all[7883] exited with preempt_count 1>
```

Debugging: Real Cases
BUG #3765

When running zram tests on arm32 qemu, while creating an ext4 filesystem on /dev/zram0,
the following crash happens, causing the rest of the lava job to time out.

This is currently happening every time on mainline and 4.16.

```
[ 1506.115245] Unable to handle kernel NULL pointer dereference at virtual address 00000000
[ 1506.116621] pgd = 0b5cc492
[ 1506.118508] Internal error: Oops: 207 [#1] SMP ARM
[ 1506.121559] CPU: 1 PID: 1883 Comm: mkfs.ext4 Not tainted 4.16.4-rc2 #1
...
[ 1506.161770] [<c0621620>] (zs_map_object) from [<bf2245fc>] (zram_bvec_rw.constprop.3+0x39c/0x660 [zram])
[ 1506.162723] [<bf2245fc>] (zram_bvec_rw.constprop.3 [zram]) from [<bf224a44>] (zram_make_request+0x184/0x374 [zram])
[ 1506.163656] [<bf224a44>] (zram_make_request [zram]) from [<c0831f80>] (generic_make_request+0x100/0x27c)
[ 1506.164487] [<c0831f80>] (generic_make_request) from [<c08321a4>] (submit_bio+0xa8/0x178)
[ 1506.165216] [<c08321a4>] (submit_bio) from [<c066d488>] (submit_bh_wbc.constprop.17+0x160/0x190)
[ 1506.165979] [<c066d488>] (submit_bh_wbc.constprop.17) from [<c066ff74>] (__block_write_full_page+0x2bc/0x538)
[ 1506.166750] [<c066ff74>] (__block_write_full_page) from [<c06704cc>] (block_write_full_page+0x168/0x170)
[ 1506.167712] [<c06704cc>] (block_write_full_page) from [<c0671c48>] (blkdev_writepage+0x24/0x28)
[ 1506.168443] [<c0671c48>] (blkdev_writepage) from [<c05b3bbc>] (__writepage+0x24/0x5c)
[ 1506.169018] [<c05b3bbc>] (__writepage) from [<c05b4518>] (write_cache_pages+0x228/0x628)
[ 1506.169772] [<c05b4518>] (write_cache_pages) from [<c05b497c>] (generic_writepages+0x64/0x90)
[ 1506.170356] [<c05b497c>] (generic_writepages) from [<c0671bf8>] (blkdev_writepages+0x18/0x1c)
[ 1506.170964] [<c0671bf8>] (blkdev_writepages) from [<c05b7040>] (do_writepages+0x3c/0xa8)
[ 1506.171524] [<c05b7040>] (do_writepages) from [<c05a4d00>] (__filemap_fdatawrite_range+0xa4/0xd8)
[ 1506.172125] [<c05a4d00>] (__filemap_fdatawrite_range) from [<c05a4f74>] (file_write_and_wait_range+0x4c/0xb4)
[ 1506.172771] [<c05a4f74>] (file_write_and_wait_range) from [<c0670e18>] (blkdev_fsync+0x2c/0x5c)
[ 1506.173359] [<c0670e18>] (blkdev_fsync) from [<c06670ec>] (vfs_fsync_range+0x4c/0xb0)
[ 1506.173906] [<c06670ec>] (vfs_fsync_range) from [<c06671d8>] (do_fsync+0x4c/0x74)
[ 1506.174508] [<c06671d8>] (do_fsync) from [<c06674e4>] (SyS_fsync+0x1c/0x20)
[ 1506.175071] [<c06674e4>] (SyS_fsync) from [<c0401000>] (ret_fast_syscall+0x0/0x28)
```