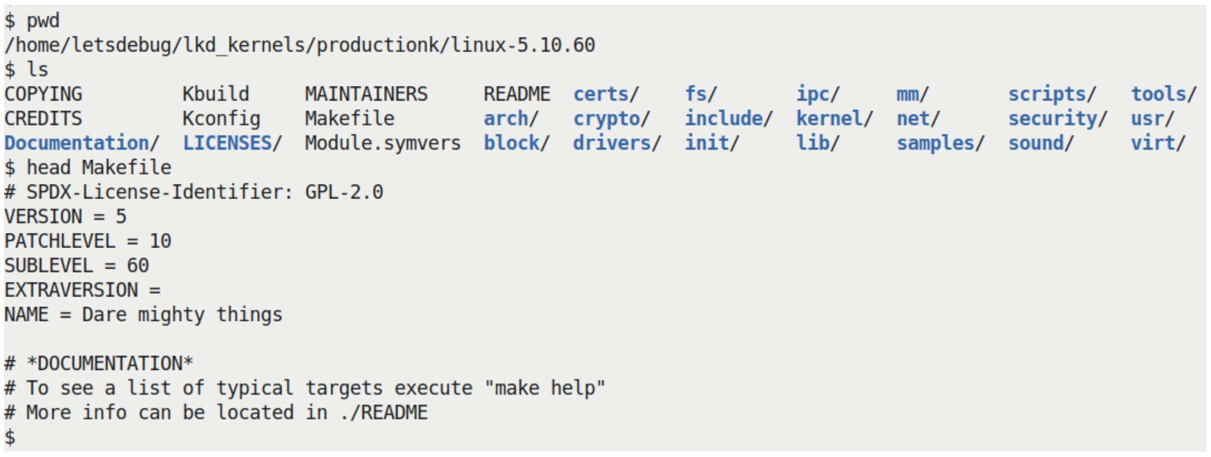
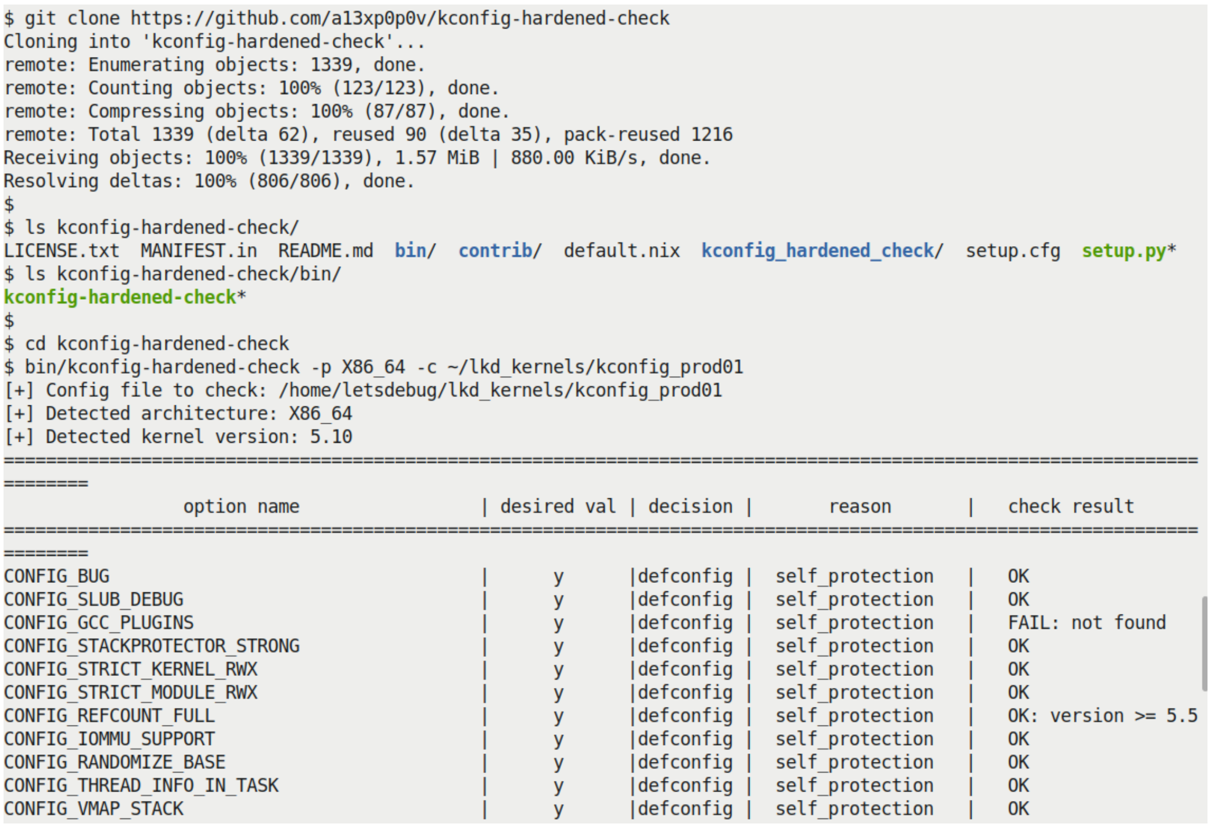
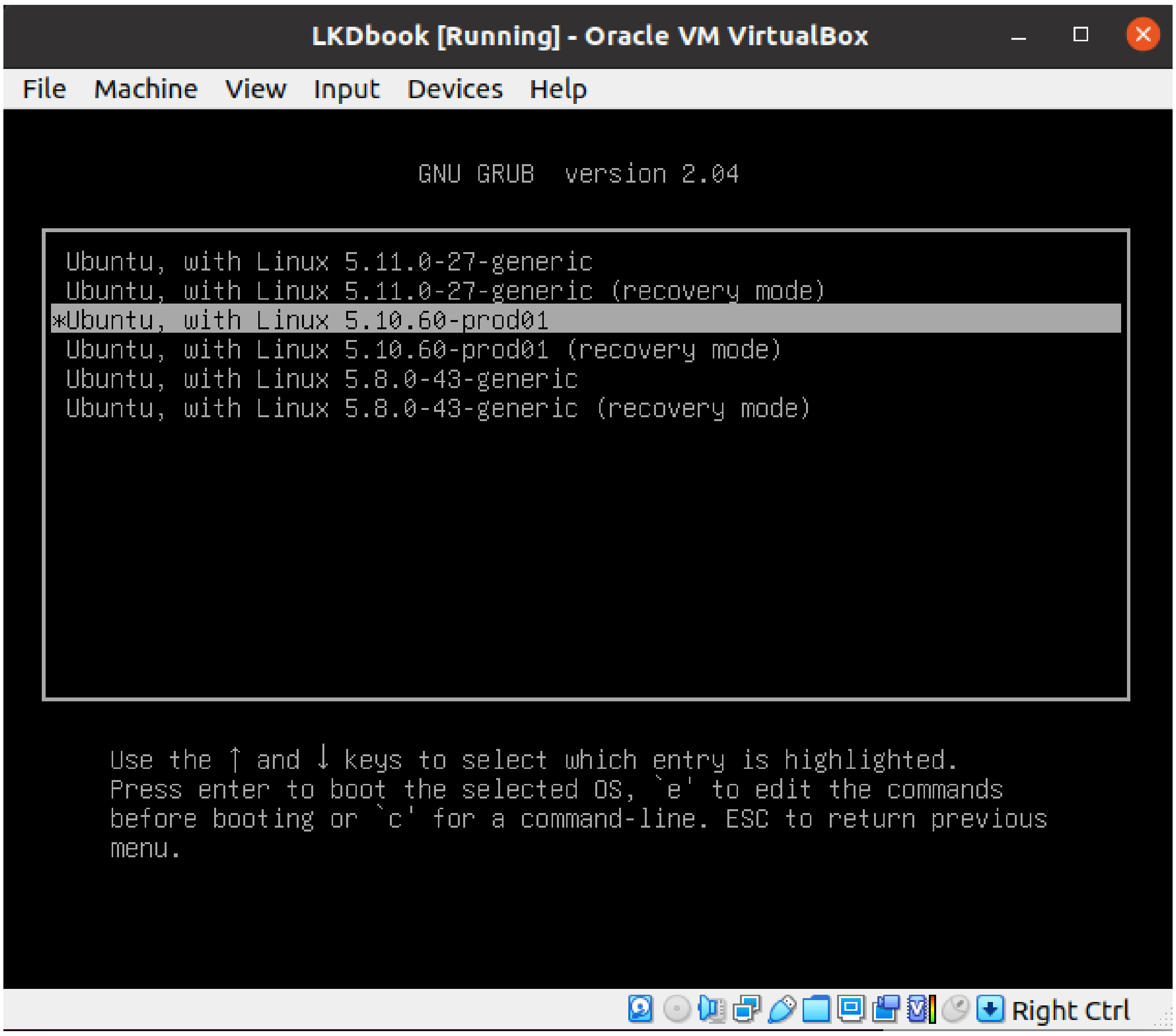
Chapter 1: A General Introduction to Debugging Software



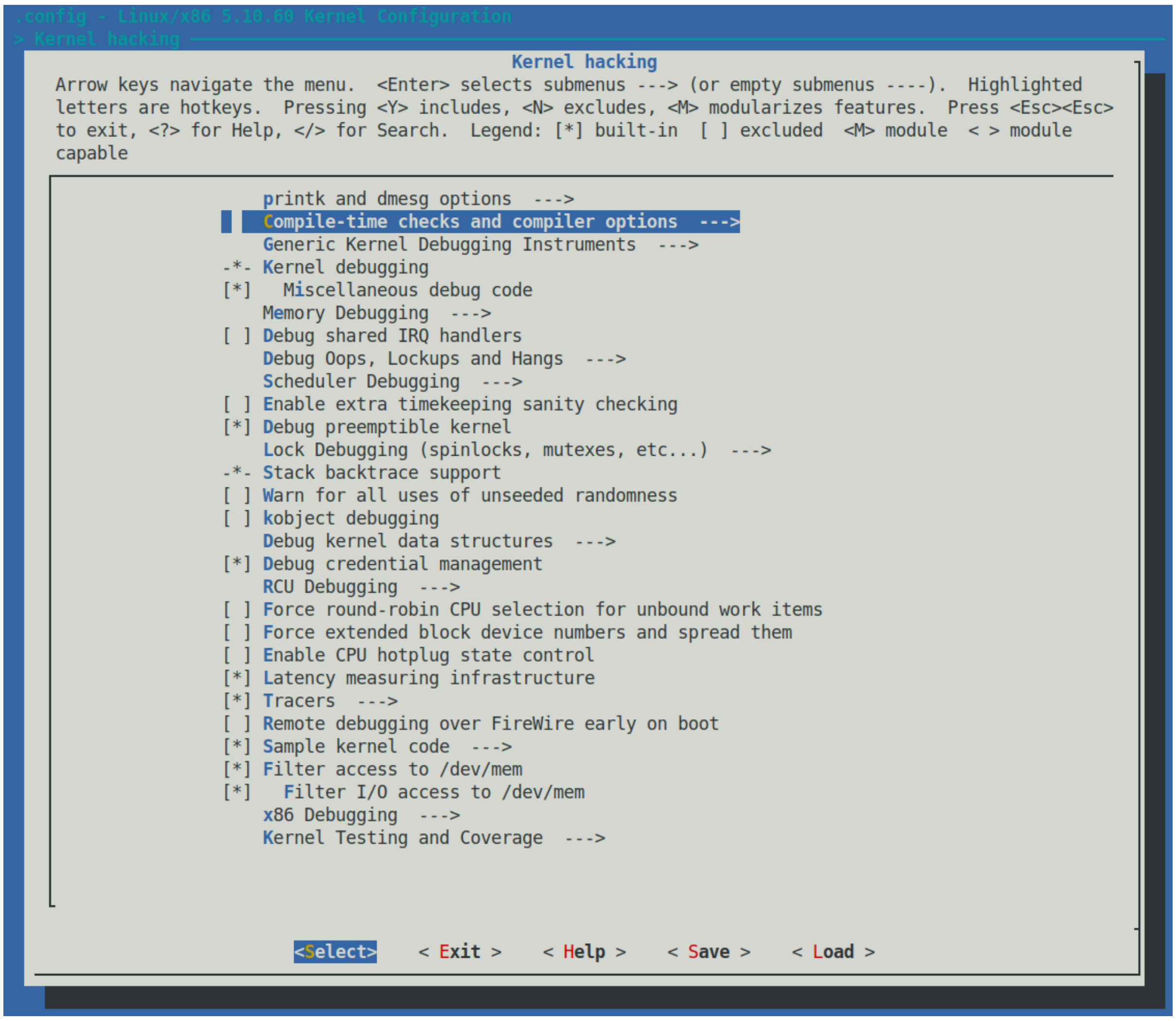
```
$ pwd
/home/letsdebug/lkd_kernels/productionk/linux-5.10.60
$ ls
COPYING         Kbuild      MAINTAINERS     README  certs/   fs/      ipc/     mm/       scripts/   tools/
CREDITS         Kconfig     Makefile        arch/   crypto/  include/ kernel/  net/      security/  usr/
Documentation/  LICENSES/   Module.symvers  block/  drivers/ init/    lib/     samples/  sound/     virt/
$ head Makefile
# SPDX-License-Identifier: GPL-2.0
VERSION = 5
PATCHLEVEL = 10
SUBLEVEL = 60
EXTRAVERSION =
NAME = Dare mighty things

# *DOCUMENTATION*
# To see a list of typical targets execute "make help"
# More info can be located in ./README
$
```



```
$ git clone https://github.com/a13xp0p0v/kconfig-hardened-check
Cloning into 'kconfig-hardened-check'...
remote: Enumerating objects: 1339, done.
remote: Counting objects: 100% (123/123), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 1339 (delta 62), reused 90 (delta 35), pack-reused 1216
Receiving objects: 100% (1339/1339), 1.57 MiB | 880.00 KiB/s, done.
Resolving deltas: 100% (806/806), done.
$
$ ls kconfig-hardened-check/
LICENSE.txt  MANIFEST.in  README.md  bin/  contrib/  default.nix  kconfig_hardened_check/  setup.cfg  setup.py*
$ ls kconfig-hardened-check/bin/
kconfig-hardened-check*
$
$ cd kconfig-hardened-check
$ bin/kconfig-hardened-check -p X86_64 -c ~/lkd_kernels/kconfig_prod01
[+] Config file to check: /home/letsdebug/lkd_kernels/kconfig_prod01
[+] Detected architecture: X86_64
[+] Detected kernel version: 5.10
=====================================================================================================
========
                option name                | desired val | decision |      reason      |   check result
=====================================================================================================
========
CONFIG_BUG                                 |      y      |defconfig |  self_protection  |   OK
CONFIG_SLUB_DEBUG                          |      y      |defconfig |  self_protection  |   OK
CONFIG_GCC_PLUGINS                         |      y      |defconfig |  self_protection  |   FAIL: not found
CONFIG_STACKPROTECTOR_STRONG               |      y      |defconfig |  self_protection  |   OK
CONFIG_STRICT_KERNEL_RWX                   |      y      |defconfig |  self_protection  |   OK
CONFIG_STRICT_MODULE_RWX                   |      y      |defconfig |  self_protection  |   OK
CONFIG_REFCOUNT_FULL                       |      y      |defconfig |  self_protection  |   OK: version >= 5.5
CONFIG_IOMMU_SUPPORT                       |      y      |defconfig |  self_protection  |   OK
CONFIG_RANDOMIZE_BASE                      |      y      |defconfig |  self_protection  |   OK
CONFIG_THREAD_INFO_IN_TASK                 |      y      |defconfig |  self_protection  |   OK
CONFIG_VMAP_STACK                          |      y      |defconfig |  self_protection  |   OK
```


LKDbook [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
                 GNU GRUB  version 2.04

 Ubuntu, with Linux 5.11.0-27-generic
 Ubuntu, with Linux 5.11.0-27-generic (recovery mode)
*Ubuntu, with Linux 5.10.60-prod01
 Ubuntu, with Linux 5.10.60-prod01 (recovery mode)
 Ubuntu, with Linux 5.8.0-43-generic
 Ubuntu, with Linux 5.8.0-43-generic (recovery mode)

    Use the ↑ and ↓ keys to select which entry is highlighted.
    Press enter to boot the selected OS, `e' to edit the commands
    before booting or `c' for a command-line. ESC to return previous
    menu.
```



```
.config - Linux/x86 5.10.60 Kernel Configuration
> Kernel hacking ──────────────────────────────────────────────────────────
                              Kernel hacking
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted
  letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc>
  to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module
  capable

                    printk and dmesg options  --->
                    Compile-time checks and compiler options  --->
                    Generic Kernel Debugging Instruments  --->
                -*- Kernel debugging
                [*]   Miscellaneous debug code
                    Memory Debugging  --->
                [ ] Debug shared IRQ handlers
                    Debug Oops, Lockups and Hangs  --->
                    Scheduler Debugging  --->
                [ ] Enable extra timekeeping sanity checking
                [*] Debug preemptible kernel
                    Lock Debugging (spinlocks, mutexes, etc...)  --->
                -*- Stack backtrace support
                [ ] Warn for all uses of unseeded randomness
                [ ] kobject debugging
                    Debug kernel data structures  --->
                [*] Debug credential management
                    RCU Debugging  --->
                [ ] Force round-robin CPU selection for unbound work items
                [ ] Force extended block device numbers and spread them
                [ ] Enable CPU hotplug state control
                [*] Latency measuring infrastructure
                [*] Tracers  --->
                [ ] Remote debugging over FireWire early on boot
                [*] Sample kernel code  --->
                [*] Filter access to /dev/mem
                [*]   Filter I/O access to /dev/mem
                    x86 Debugging  --->
                    Kernel Testing and Coverage  --->

                   <Select>    < Exit >    < Help >    < Save >    < Load >
```

empirical

/ɛmˈpɪrɪk(ə)l,ɪmˈpɪrɪk(ə)l/

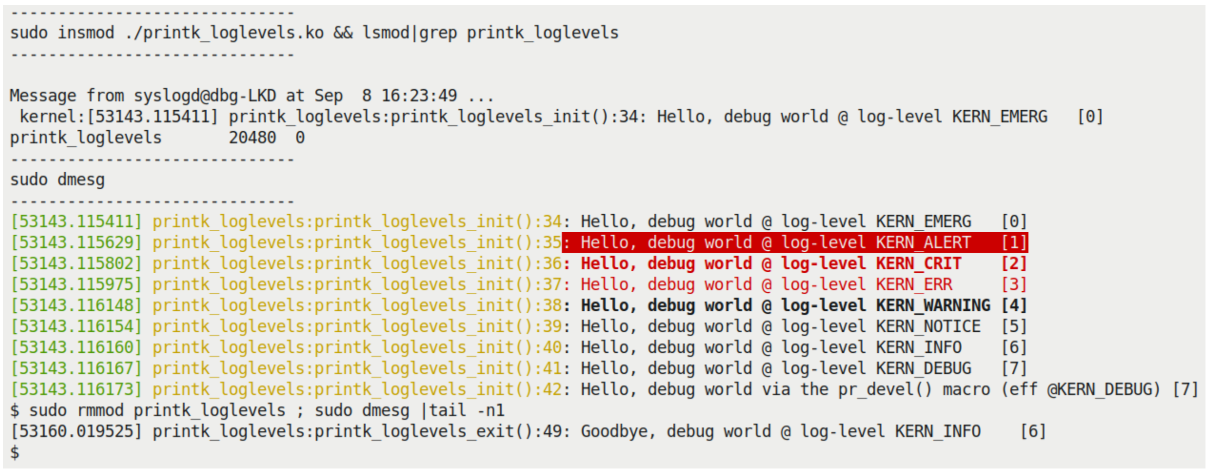
adjective

based on, concerned with, or verifiable by observation or experience rather than theory or pure logic.

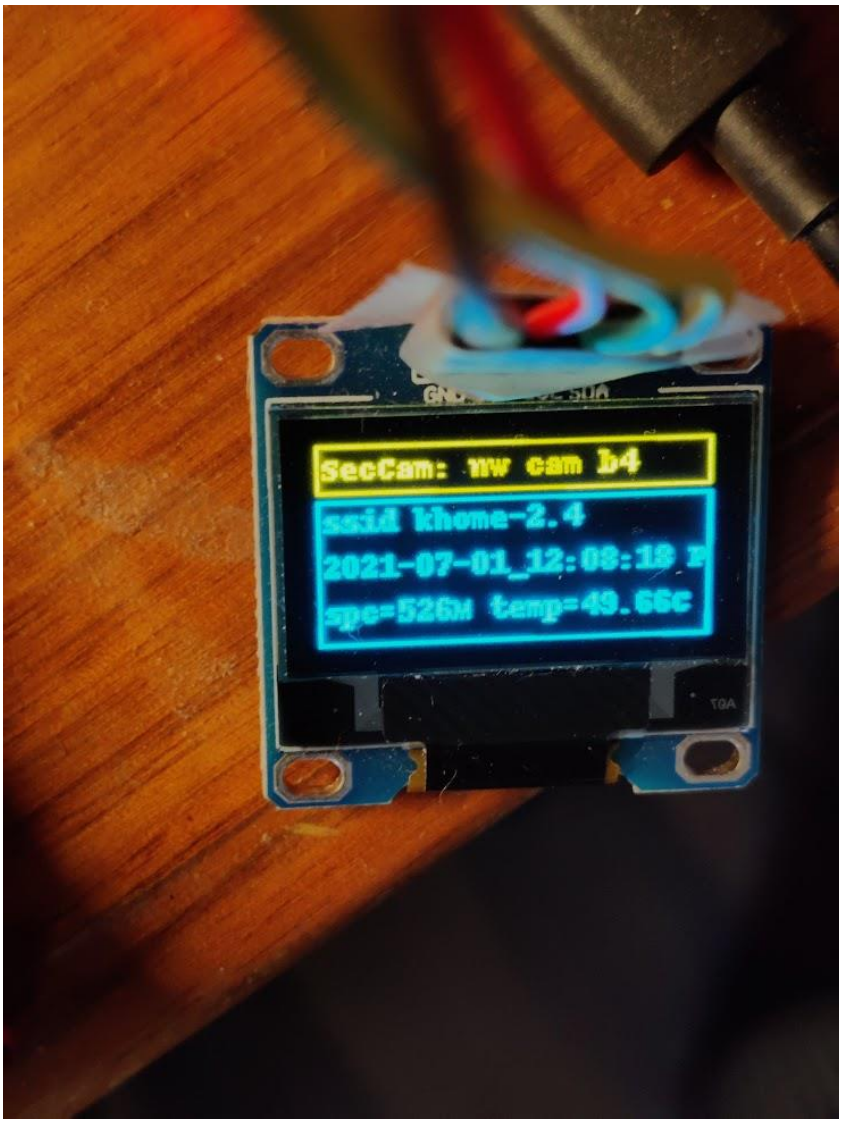
Chapter 2: Approaches to Kernel Debugging

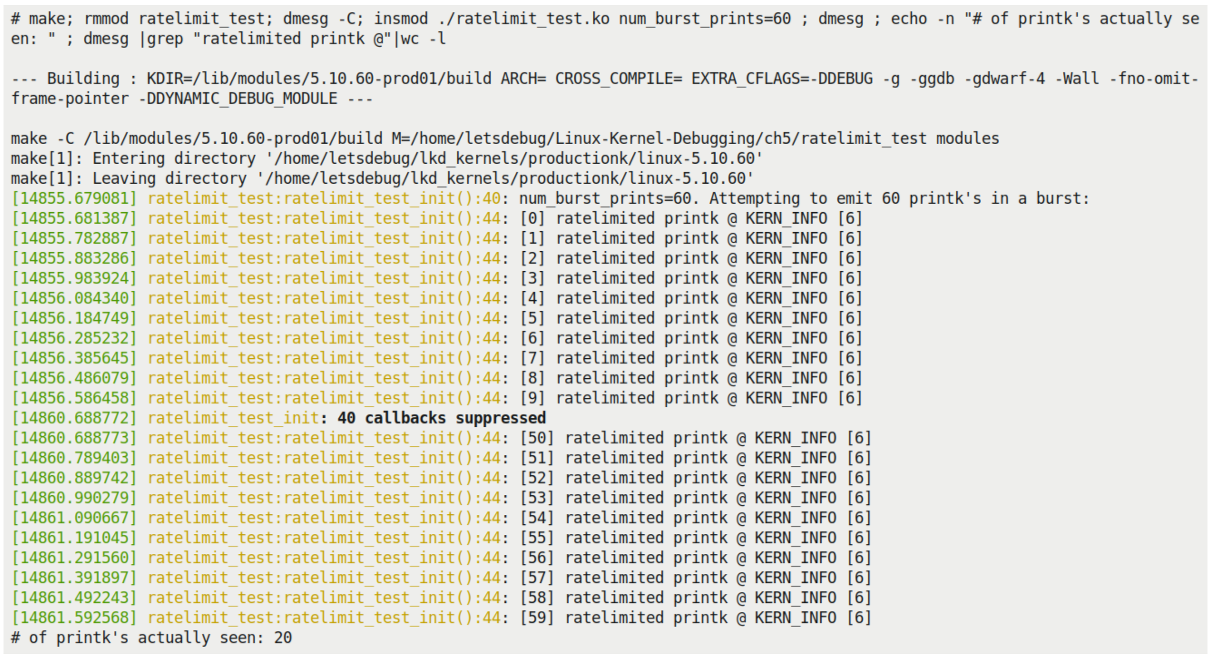
No Images

Chapter 3: Debug via Instrumentation – printk and Friends

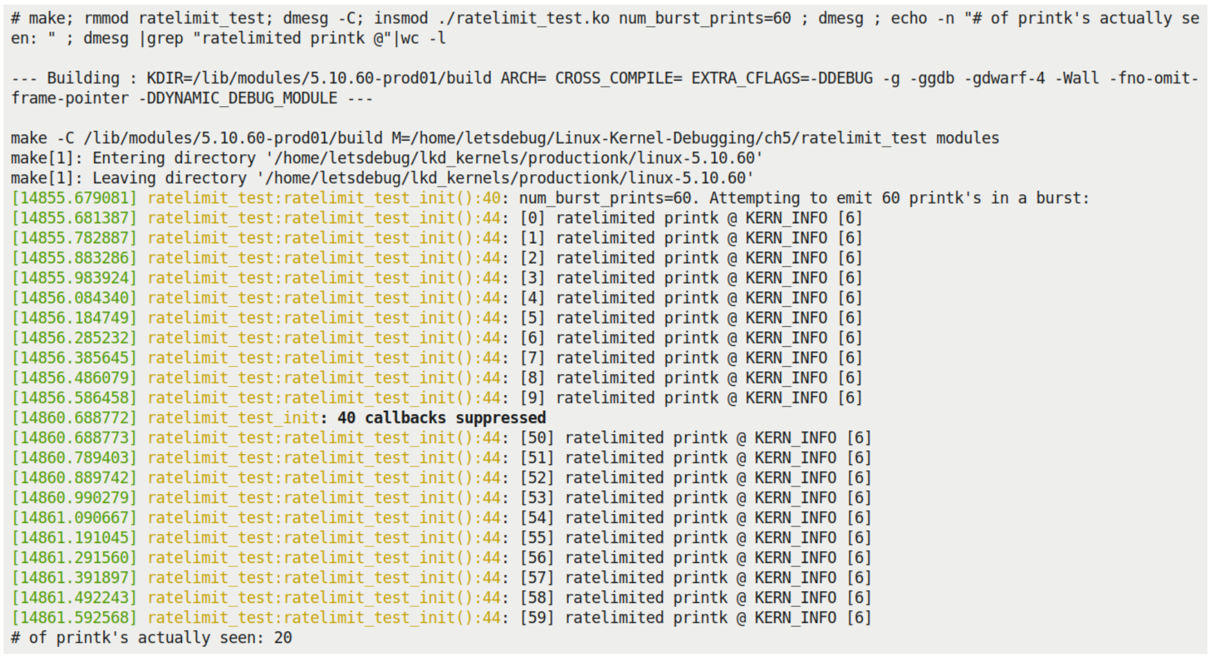


```
------------------------------
sudo insmod ./printk_loglevels.ko && lsmod|grep printk_loglevels
------------------------------

Message from syslogd@dbg-LKD at Sep  8 16:23:49 ...
 kernel:[53143.115411] printk_loglevels:printk_loglevels_init():34: Hello, debug world @ log-level KERN_EMERG   [0]
printk_loglevels       20480  0
------------------------------
sudo dmesg
------------------------------
[53143.115411] printk_loglevels:printk_loglevels_init():34: Hello, debug world @ log-level KERN_EMERG   [0]
[53143.115629] printk_loglevels:printk_loglevels_init():35: Hello, debug world @ log-level KERN_ALERT   [1]
[53143.115802] printk_loglevels:printk_loglevels_init():36: Hello, debug world @ log-level KERN_CRIT    [2]
[53143.115975] printk_loglevels:printk_loglevels_init():37: Hello, debug world @ log-level KERN_ERR     [3]
[53143.116148] printk_loglevels:printk_loglevels_init():38: Hello, debug world @ log-level KERN_WARNING [4]
[53143.116154] printk_loglevels:printk_loglevels_init():39: Hello, debug world @ log-level KERN_NOTICE  [5]
[53143.116160] printk_loglevels:printk_loglevels_init():40: Hello, debug world @ log-level KERN_INFO    [6]
[53143.116167] printk_loglevels:printk_loglevels_init():41: Hello, debug world @ log-level KERN_DEBUG   [7]
[53143.116173] printk_loglevels:printk_loglevels_init():42: Hello, debug world via the pr_devel() macro (eff @KERN_DEBUG) [7]
$ sudo rmmod printk_loglevels ; sudo dmesg |tail -n1
[53160.019525] printk_loglevels:printk_loglevels_exit():49: Goodbye, debug world @ log-level KERN_INFO    [6]
$
```

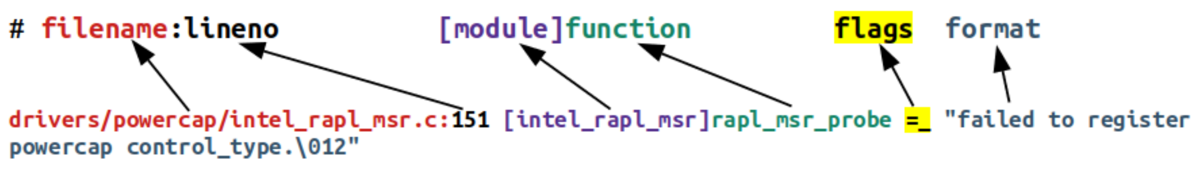




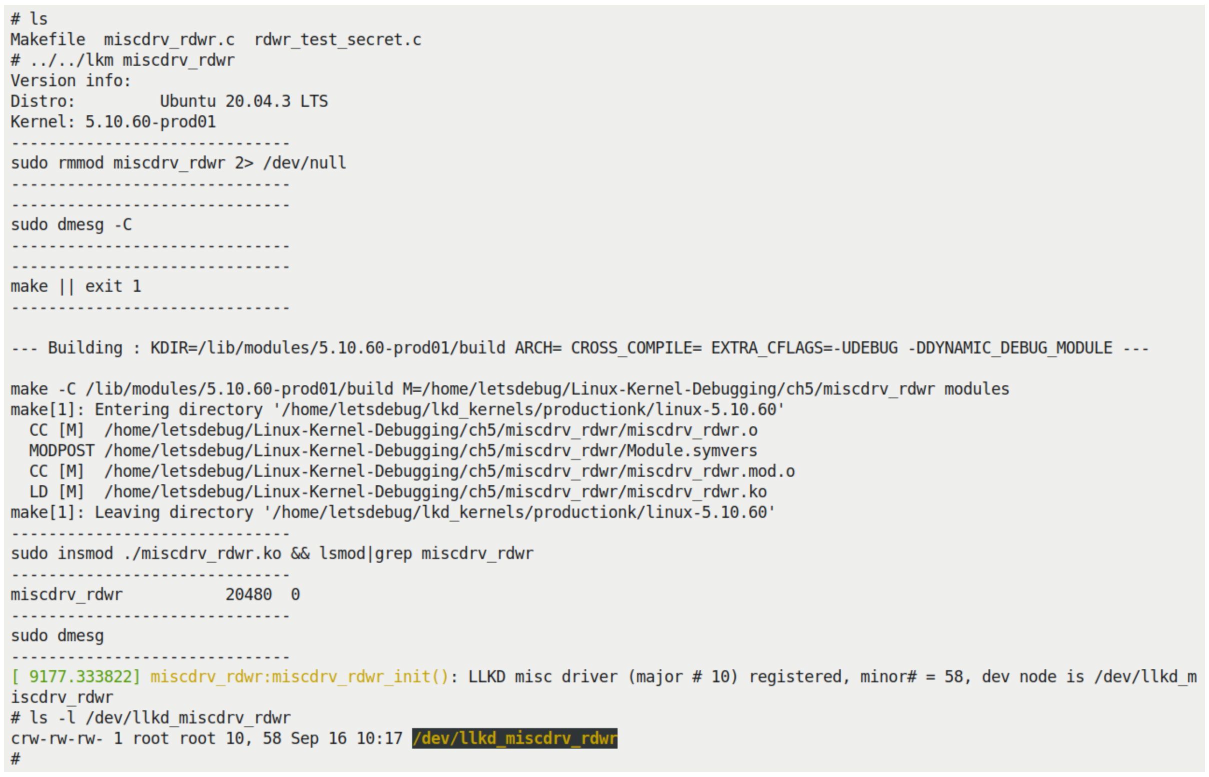
```
# make; rmmod ratelimit_test; dmesg -C; insmod ./ratelimit_test.ko num_burst_prints=60 ; dmesg ; echo -n "# of printk's actually se
en: " ; dmesg |grep "ratelimited printk @"|wc -l

--- Building : KDIR=/lib/modules/5.10.60-prod01/build ARCH= CROSS_COMPILE= EXTRA_CFLAGS=-DDEBUG -g -ggdb -gdwarf-4 -Wall -fno-omit-
frame-pointer -DDYNAMIC_DEBUG_MODULE ---

make -C /lib/modules/5.10.60-prod01/build M=/home/letsdebug/Linux-Kernel-Debugging/ch5/ratelimit_test modules
make[1]: Entering directory '/home/letsdebug/lkd_kernels/productionk/linux-5.10.60'
make[1]: Leaving directory '/home/letsdebug/lkd_kernels/productionk/linux-5.10.60'
[14855.679081] ratelimit_test:ratelimit_test_init():40: num_burst_prints=60. Attempting to emit 60 printk's in a burst:
[14855.681387] ratelimit_test:ratelimit_test_init():44: [0] ratelimited printk @ KERN_INFO [6]
[14855.782887] ratelimit_test:ratelimit_test_init():44: [1] ratelimited printk @ KERN_INFO [6]
[14855.883286] ratelimit_test:ratelimit_test_init():44: [2] ratelimited printk @ KERN_INFO [6]
[14855.983924] ratelimit_test:ratelimit_test_init():44: [3] ratelimited printk @ KERN_INFO [6]
[14856.084340] ratelimit_test:ratelimit_test_init():44: [4] ratelimited printk @ KERN_INFO [6]
[14856.184749] ratelimit_test:ratelimit_test_init():44: [5] ratelimited printk @ KERN_INFO [6]
[14856.285232] ratelimit_test:ratelimit_test_init():44: [6] ratelimited printk @ KERN_INFO [6]
[14856.385645] ratelimit_test:ratelimit_test_init():44: [7] ratelimited printk @ KERN_INFO [6]
[14856.486079] ratelimit_test:ratelimit_test_init():44: [8] ratelimited printk @ KERN_INFO [6]
[14856.586458] ratelimit_test:ratelimit_test_init():44: [9] ratelimited printk @ KERN_INFO [6]
[14860.688772] ratelimit_test_init: 40 callbacks suppressed
[14860.688773] ratelimit_test:ratelimit_test_init():44: [50] ratelimited printk @ KERN_INFO [6]
[14860.789403] ratelimit_test:ratelimit_test_init():44: [51] ratelimited printk @ KERN_INFO [6]
[14860.889742] ratelimit_test:ratelimit_test_init():44: [52] ratelimited printk @ KERN_INFO [6]
[14860.990279] ratelimit_test:ratelimit_test_init():44: [53] ratelimited printk @ KERN_INFO [6]
[14861.090667] ratelimit_test:ratelimit_test_init():44: [54] ratelimited printk @ KERN_INFO [6]
[14861.191045] ratelimit_test:ratelimit_test_init():44: [55] ratelimited printk @ KERN_INFO [6]
[14861.291560] ratelimit_test:ratelimit_test_init():44: [56] ratelimited printk @ KERN_INFO [6]
[14861.391897] ratelimit_test:ratelimit_test_init():44: [57] ratelimited printk @ KERN_INFO [6]
[14861.492243] ratelimit_test:ratelimit_test_init():44: [58] ratelimited printk @ KERN_INFO [6]
[14861.592568] ratelimit_test:ratelimit_test_init():44: [59] ratelimited printk @ KERN_INFO [6]
# of printk's actually seen: 20
```



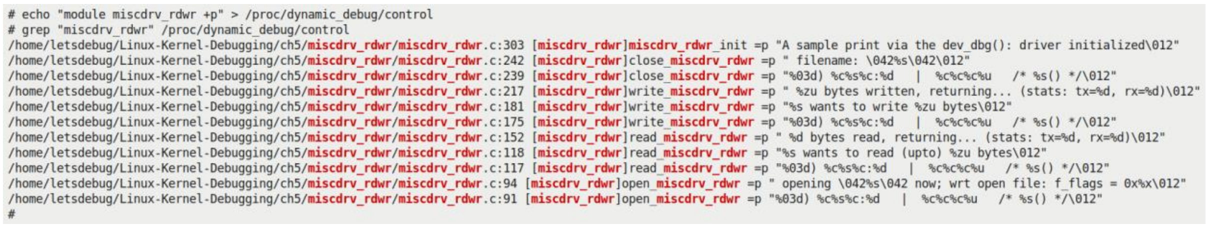
```
# make; rmmod ratelimit_test; dmesg -C; insmod ./ratelimit_test.ko num_burst_prints=60 ; dmesg ; echo -n "# of printk's actually se
en: " ; dmesg |grep "ratelimited printk @"|wc -l

--- Building : KDIR=/lib/modules/5.10.60-prod01/build ARCH= CROSS_COMPILE= EXTRA_CFLAGS=-DDEBUG -g -ggdb -gdwarf-4 -Wall -fno-omit-
frame-pointer -DDYNAMIC_DEBUG_MODULE ---

make -C /lib/modules/5.10.60-prod01/build M=/home/letsdebug/Linux-Kernel-Debugging/ch5/ratelimit_test modules
make[1]: Entering directory '/home/letsdebug/lkd_kernels/productionk/linux-5.10.60'
make[1]: Leaving directory '/home/letsdebug/lkd_kernels/productionk/linux-5.10.60'
[14855.679081] ratelimit_test:ratelimit_test_init():40: num_burst_prints=60. Attempting to emit 60 printk's in a burst:
[14855.681387] ratelimit_test:ratelimit_test_init():44: [0] ratelimited printk @ KERN_INFO [6]
[14855.782887] ratelimit_test:ratelimit_test_init():44: [1] ratelimited printk @ KERN_INFO [6]
[14855.883286] ratelimit_test:ratelimit_test_init():44: [2] ratelimited printk @ KERN_INFO [6]
[14855.983924] ratelimit_test:ratelimit_test_init():44: [3] ratelimited printk @ KERN_INFO [6]
[14856.084340] ratelimit_test:ratelimit_test_init():44: [4] ratelimited printk @ KERN_INFO [6]
[14856.184749] ratelimit_test:ratelimit_test_init():44: [5] ratelimited printk @ KERN_INFO [6]
[14856.285232] ratelimit_test:ratelimit_test_init():44: [6] ratelimited printk @ KERN_INFO [6]
[14856.385645] ratelimit_test:ratelimit_test_init():44: [7] ratelimited printk @ KERN_INFO [6]
[14856.486079] ratelimit_test:ratelimit_test_init():44: [8] ratelimited printk @ KERN_INFO [6]
[14856.586458] ratelimit_test:ratelimit_test_init():44: [9] ratelimited printk @ KERN_INFO [6]
[14860.688772] ratelimit_test_init: 40 callbacks suppressed
[14860.688773] ratelimit_test:ratelimit_test_init():44: [50] ratelimited printk @ KERN_INFO [6]
[14860.789403] ratelimit_test:ratelimit_test_init():44: [51] ratelimited printk @ KERN_INFO [6]
[14860.889742] ratelimit_test:ratelimit_test_init():44: [52] ratelimited printk @ KERN_INFO [6]
[14860.990279] ratelimit_test:ratelimit_test_init():44: [53] ratelimited printk @ KERN_INFO [6]
[14861.090667] ratelimit_test:ratelimit_test_init():44: [54] ratelimited printk @ KERN_INFO [6]
[14861.191045] ratelimit_test:ratelimit_test_init():44: [55] ratelimited printk @ KERN_INFO [6]
[14861.291560] ratelimit_test:ratelimit_test_init():44: [56] ratelimited printk @ KERN_INFO [6]
[14861.391897] ratelimit_test:ratelimit_test_init():44: [57] ratelimited printk @ KERN_INFO [6]
[14861.492243] ratelimit_test:ratelimit_test_init():44: [58] ratelimited printk @ KERN_INFO [6]
[14861.592568] ratelimit_test:ratelimit_test_init():44: [59] ratelimited printk @ KERN_INFO [6]
# of printk's actually seen: 20
```



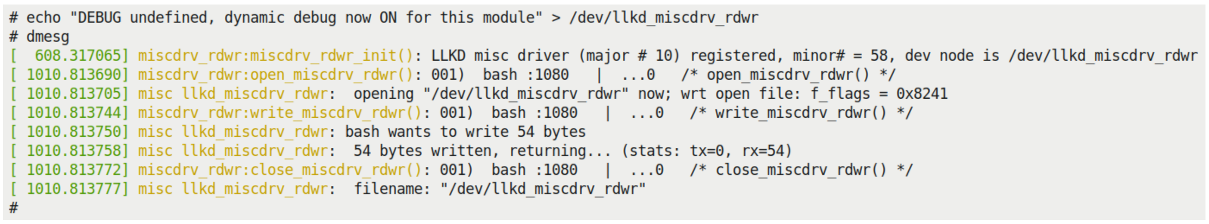
# filename:lineno
[module]function
flags
format
drivers/powercap/intel_rapl_msr.c:151 [intel_rapl_msr]rapl_msr_probe =_ "failed to register powercap control_type.\012"



```
# ls
Makefile  miscdrv_rdwr.c  rdwr_test_secret.c
# ../../lkm miscdrv_rdwr
Version info:
Distro:         Ubuntu 20.04.3 LTS
Kernel: 5.10.60-prod01
------------------------------
sudo rmmod miscdrv_rdwr 2> /dev/null
------------------------------
------------------------------
sudo dmesg -C
------------------------------
------------------------------
make || exit 1
------------------------------

--- Building : KDIR=/lib/modules/5.10.60-prod01/build ARCH= CROSS_COMPILE= EXTRA_CFLAGS=-UDEBUG -DDYNAMIC_DEBUG_MODULE ---

make -C /lib/modules/5.10.60-prod01/build M=/home/letsdebug/Linux-Kernel-Debugging/ch5/miscdrv_rdwr modules
make[1]: Entering directory '/home/letsdebug/lkd_kernels/productionk/linux-5.10.60'
  CC [M]  /home/letsdebug/Linux-Kernel-Debugging/ch5/miscdrv_rdwr/miscdrv_rdwr.o
  MODPOST /home/letsdebug/Linux-Kernel-Debugging/ch5/miscdrv_rdwr/Module.symvers
  CC [M]  /home/letsdebug/Linux-Kernel-Debugging/ch5/miscdrv_rdwr/miscdrv_rdwr.mod.o
  LD [M]  /home/letsdebug/Linux-Kernel-Debugging/ch5/miscdrv_rdwr/miscdrv_rdwr.ko
make[1]: Leaving directory '/home/letsdebug/lkd_kernels/productionk/linux-5.10.60'
------------------------------
sudo insmod ./miscdrv_rdwr.ko && lsmod|grep miscdrv_rdwr
------------------------------
miscdrv_rdwr           20480  0
------------------------------
sudo dmesg
------------------------------
[ 9177.333822] miscdrv_rdwr:miscdrv_rdwr_init(): LLKD misc driver (major # 10) registered, minor# = 58, dev node is /dev/llkd_m
iscdrv_rdwr
# ls -l /dev/llkd_miscdrv_rdwr
crw-rw-rw- 1 root root 10, 58 Sep 16 10:17 /dev/llkd_miscdrv_rdwr
#
```

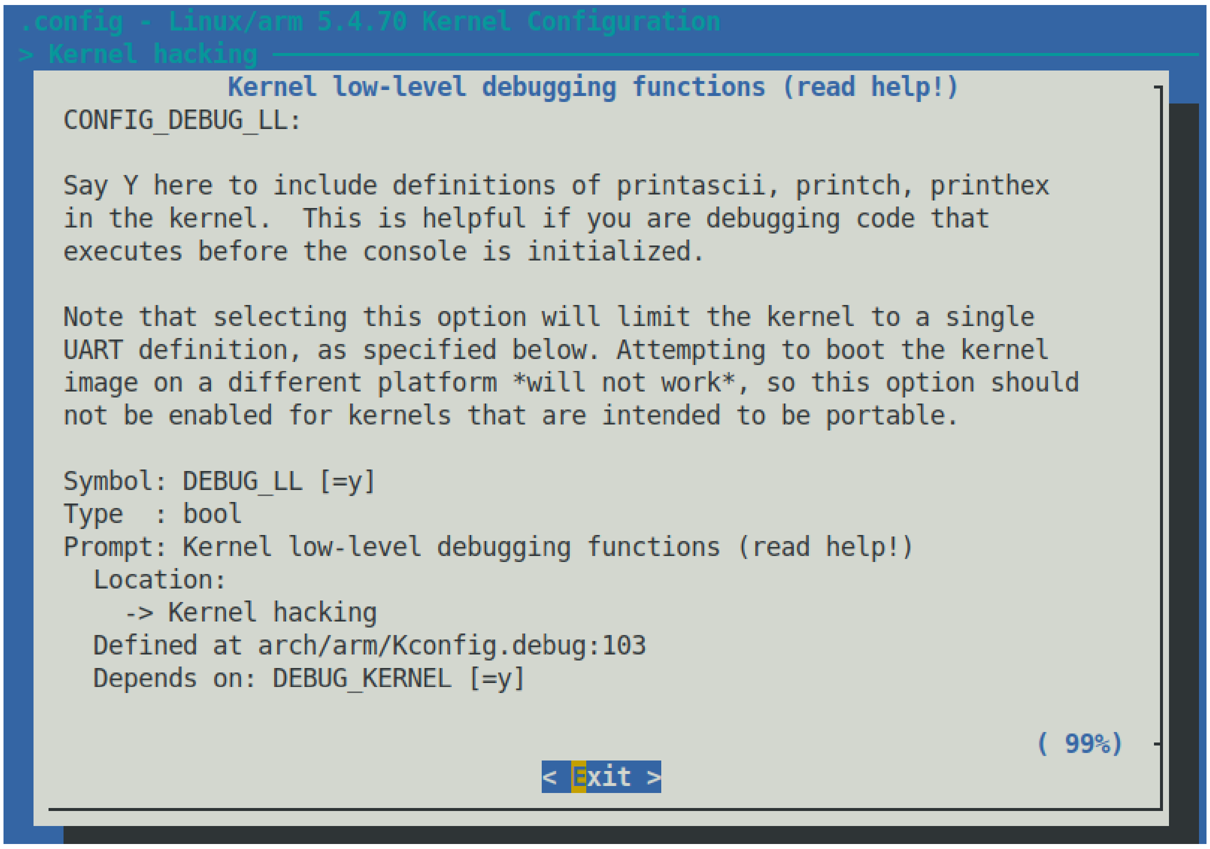


```
# echo "module miscdrv_rdwr +p" > /proc/dynamic_debug/control
# grep "miscdrv_rdwr" /proc/dynamic_debug/control
/home/letsdebug/Linux-Kernel-Debugging/ch5/miscdrv_rdwr/miscdrv_rdwr.c:303 [miscdrv_rdwr]miscdrv_rdwr_init =p "A sample print via the dev_dbg(): driver initialized\012"
/home/letsdebug/Linux-Kernel-Debugging/ch5/miscdrv_rdwr/miscdrv_rdwr.c:242 [miscdrv_rdwr]close_miscdrv_rdwr =p " filename: \042%s\042\012"
/home/letsdebug/Linux-Kernel-Debugging/ch5/miscdrv_rdwr/miscdrv_rdwr.c:239 [miscdrv_rdwr]close_miscdrv_rdwr =p "%03d) %c%s%c:%d   |  %c%c%c%u   /* %s() */\012"
/home/letsdebug/Linux-Kernel-Debugging/ch5/miscdrv_rdwr/miscdrv_rdwr.c:217 [miscdrv_rdwr]write_miscdrv_rdwr =p " %zu bytes written, returning... (stats: tx=%d, rx=%d)\012"
/home/letsdebug/Linux-Kernel-Debugging/ch5/miscdrv_rdwr/miscdrv_rdwr.c:181 [miscdrv_rdwr]write_miscdrv_rdwr =p "%s wants to write %zu bytes\012"
/home/letsdebug/Linux-Kernel-Debugging/ch5/miscdrv_rdwr/miscdrv_rdwr.c:175 [miscdrv_rdwr]write_miscdrv_rdwr =p "%03d) %c%s%c:%d   |  %c%c%c%u   /* %s() */\012"
/home/letsdebug/Linux-Kernel-Debugging/ch5/miscdrv_rdwr/miscdrv_rdwr.c:152 [miscdrv_rdwr]read_miscdrv_rdwr =p " %d bytes read, returning... (stats: tx=%d, rx=%d)\012"
/home/letsdebug/Linux-Kernel-Debugging/ch5/miscdrv_rdwr/miscdrv_rdwr.c:118 [miscdrv_rdwr]read_miscdrv_rdwr =p "%s wants to read (upto) %zu bytes\012"
/home/letsdebug/Linux-Kernel-Debugging/ch5/miscdrv_rdwr/miscdrv_rdwr.c:117 [miscdrv_rdwr]read_miscdrv_rdwr =p "%03d) %c%s%c:%d   |  %c%c%c%u   /* %s() */\012"
/home/letsdebug/Linux-Kernel-Debugging/ch5/miscdrv_rdwr/miscdrv_rdwr.c:94 [miscdrv_rdwr]open_miscdrv_rdwr =p " opening \042%s\042 now; wrt open file: f_flags = 0x%x\012"
/home/letsdebug/Linux-Kernel-Debugging/ch5/miscdrv_rdwr/miscdrv_rdwr.c:91 [miscdrv_rdwr]open_miscdrv_rdwr =p "%03d) %c%s%c:%d   |  %c%c%c%u   /* %s() */\012"
#
```



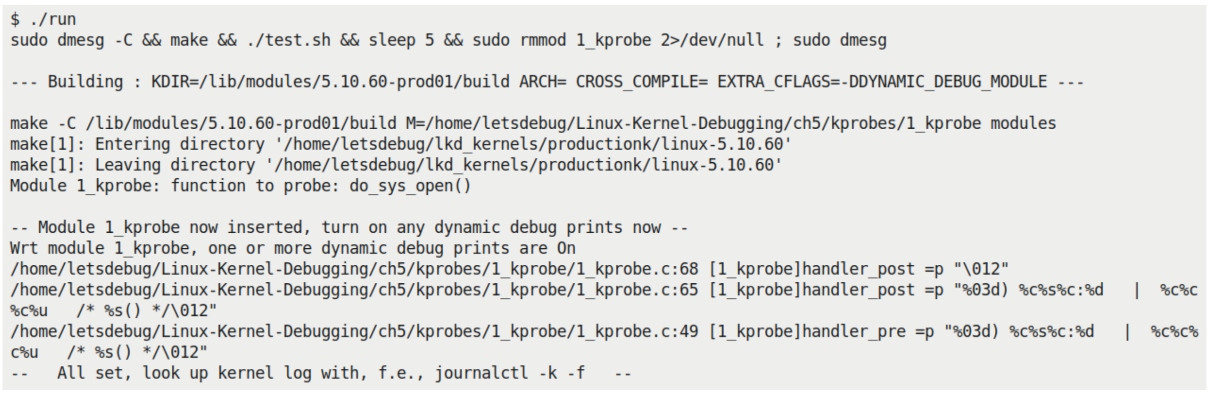
```
# echo "DEBUG undefined, dynamic debug now ON for this module" > /dev/llkd_miscdrv_rdwr
# dmesg
[  608.317065] miscdrv_rdwr:miscdrv_rdwr_init(): LLKD misc driver (major # 10) registered, minor# = 58, dev node is /dev/llkd_miscdrv_rdwr
[ 1010.813690] miscdrv_rdwr:open_miscdrv_rdwr(): 001)  bash :1080   |  ...0   /* open_miscdrv_rdwr() */
[ 1010.813705] misc llkd_miscdrv_rdwr:  opening "/dev/llkd_miscdrv_rdwr" now; wrt open file: f_flags = 0x8241
[ 1010.813744] miscdrv_rdwr:write_miscdrv_rdwr(): 001)  bash :1080   |  ...0   /* write_miscdrv_rdwr() */
[ 1010.813750] misc llkd_miscdrv_rdwr: bash wants to write 54 bytes
[ 1010.813758] misc llkd_miscdrv_rdwr:  54 bytes written, returning... (stats: tx=0, rx=54)
[ 1010.813772] miscdrv_rdwr:close_miscdrv_rdwr(): 001)  bash :1080   |  ...0   /* close_miscdrv_rdwr() */
[ 1010.813777] misc llkd_miscdrv_rdwr:  filename: "/dev/llkd_miscdrv_rdwr"
#
```

```
config EARLY_PRINTK
    bool "Early printk" if EXPERT
    default y
    help
      Write kernel log output directly into the VGA buffer or to a serial
      port.

      This is useful for kernel debugging when your machine crashes very
      early before the console code is initialized. For normal operation
      it is not recommended because it looks ugly and doesn't cooperate
      with klogd/syslogd or the X server. You should normally say N here,
      unless you want to debug such a crash.
```

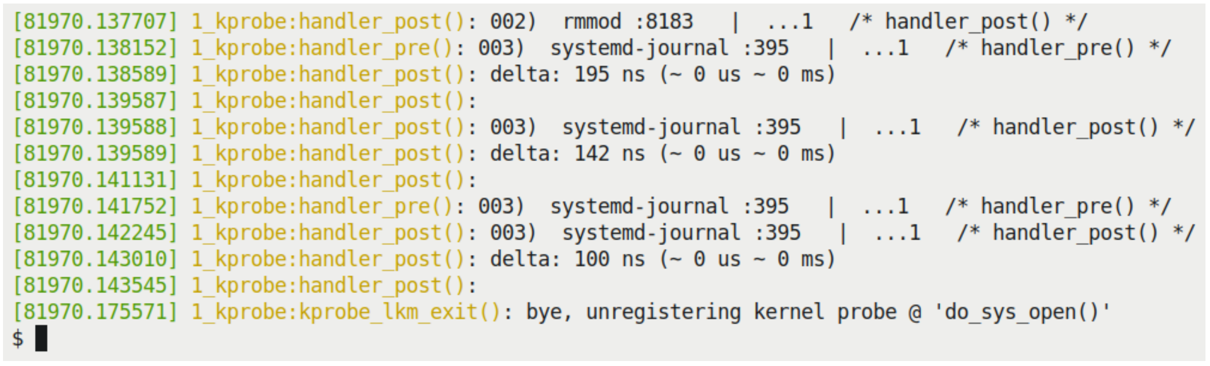


```
.config - Linux/arm 5.4.70 Kernel Configuration
> Kernel hacking
          Kernel low-level debugging functions (read help!)
CONFIG_DEBUG_LL:

Say Y here to include definitions of printascii, printch, printhex
in the kernel.  This is helpful if you are debugging code that
executes before the console is initialized.

Note that selecting this option will limit the kernel to a single
UART definition, as specified below. Attempting to boot the kernel
image on a different platform *will not work*, so this option should
not be enabled for kernels that are intended to be portable.

Symbol: DEBUG_LL [=y]
Type  : bool
Prompt: Kernel low-level debugging functions (read help!)
  Location:
    -> Kernel hacking
  Defined at arch/arm/Kconfig.debug:103
  Depends on: DEBUG_KERNEL [=y]

                                                          ( 99%)
```

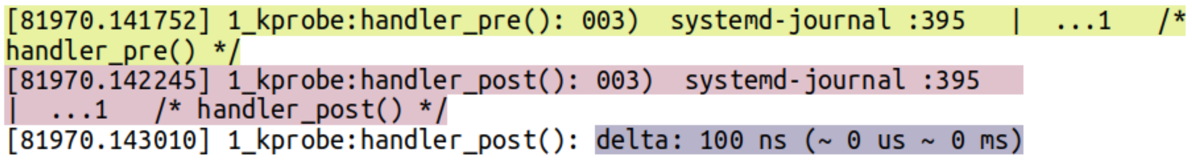
Chapter 4: Debug via Instrumentation – Kprobes



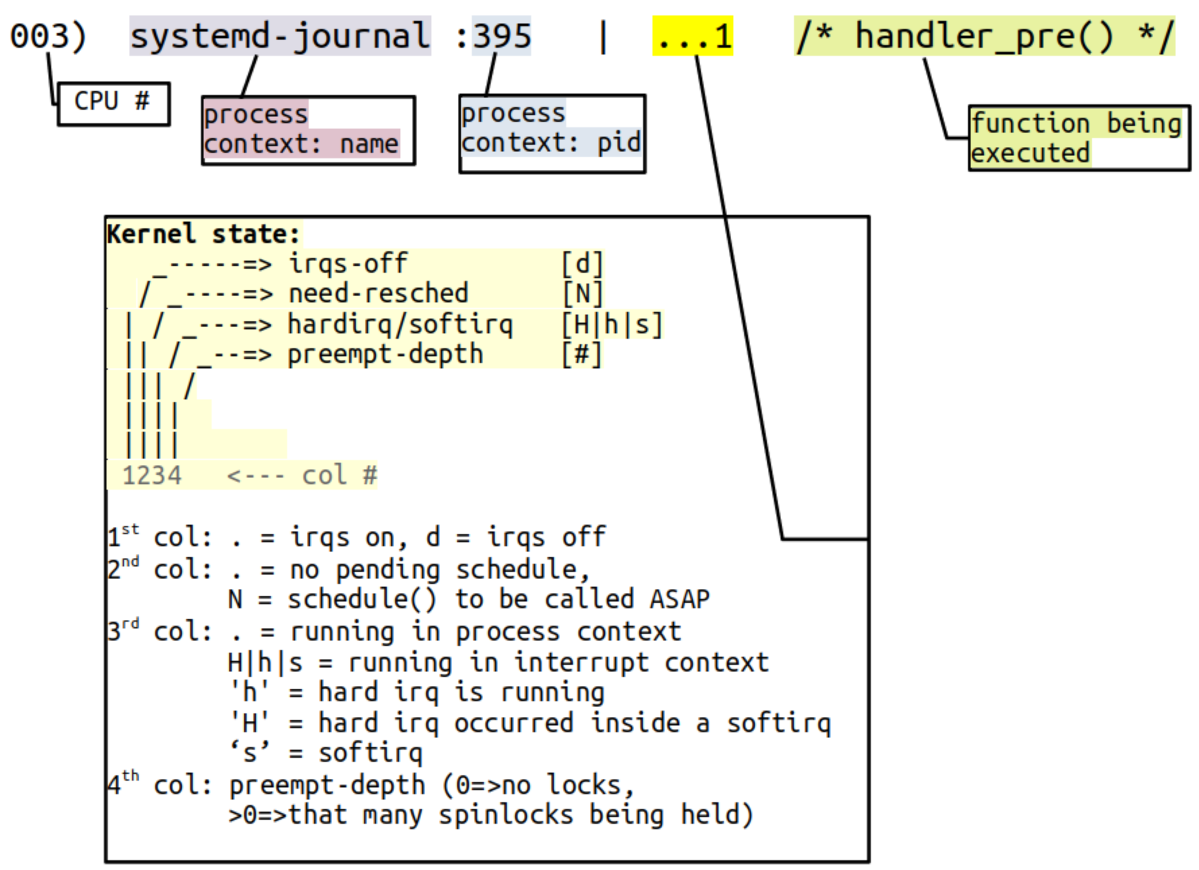
```
$ ./run
sudo dmesg -C && make && ./test.sh && sleep 5 && sudo rmmod 1_kprobe 2>/dev/null ; sudo dmesg

--- Building : KDIR=/lib/modules/5.10.60-prod01/build ARCH= CROSS_COMPILE= EXTRA_CFLAGS=-DDYNAMIC_DEBUG_MODULE ---

make -C /lib/modules/5.10.60-prod01/build M=/home/letsdebug/Linux-Kernel-Debugging/ch5/kprobes/1_kprobe modules
make[1]: Entering directory '/home/letsdebug/lkd_kernels/productionk/linux-5.10.60'
make[1]: Leaving directory '/home/letsdebug/lkd_kernels/productionk/linux-5.10.60'
Module 1_kprobe: function to probe: do_sys_open()

-- Module 1_kprobe now inserted, turn on any dynamic debug prints now --
Wrt module 1_kprobe, one or more dynamic debug prints are On
/home/letsdebug/Linux-Kernel-Debugging/ch5/kprobes/1_kprobe/1_kprobe.c:68 [1_kprobe]handler_post =p "\012"
/home/letsdebug/Linux-Kernel-Debugging/ch5/kprobes/1_kprobe/1_kprobe.c:65 [1_kprobe]handler_post =p "%03d) %c%s%c:%d   |  %c%c
%c%u   /* %s() */\012"
/home/letsdebug/Linux-Kernel-Debugging/ch5/kprobes/1_kprobe/1_kprobe.c:49 [1_kprobe]handler_pre =p "%03d) %c%s%c:%d   |  %c%c%
c%u   /* %s() */\012"
--   All set, look up kernel log with, f.e., journalctl -k -f   --
```



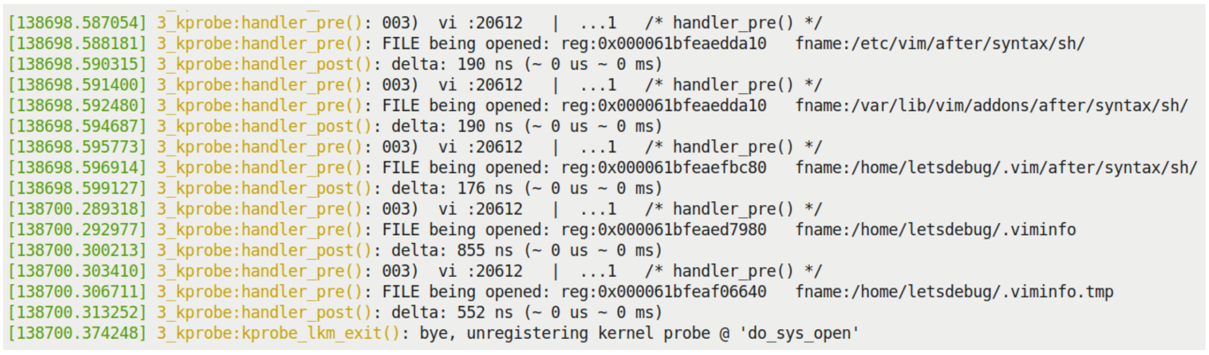
```
[81970.137707] 1_kprobe:handler_post(): 002)  rmmod :8183   |  ...1   /* handler_post() */
[81970.138152] 1_kprobe:handler_pre(): 003)  systemd-journal :395   |  ...1   /* handler_pre() */
[81970.138589] 1_kprobe:handler_post(): delta: 195 ns (~ 0 us ~ 0 ms)
[81970.139587] 1_kprobe:handler_post():
[81970.139588] 1_kprobe:handler_post(): 003)  systemd-journal :395   |  ...1   /* handler_post() */
[81970.139589] 1_kprobe:handler_post(): delta: 142 ns (~ 0 us ~ 0 ms)
[81970.141131] 1_kprobe:handler_post():
[81970.141752] 1_kprobe:handler_pre(): 003)  systemd-journal :395   |  ...1   /* handler_pre() */
[81970.142245] 1_kprobe:handler_post(): 003)  systemd-journal :395   |  ...1   /* handler_post() */
[81970.143010] 1_kprobe:handler_post(): delta: 100 ns (~ 0 us ~ 0 ms)
[81970.143545] 1_kprobe:handler_post():
[81970.175571] 1_kprobe:kprobe_lkm_exit(): bye, unregistering kernel probe @ 'do_sys_open()'
$ 
```



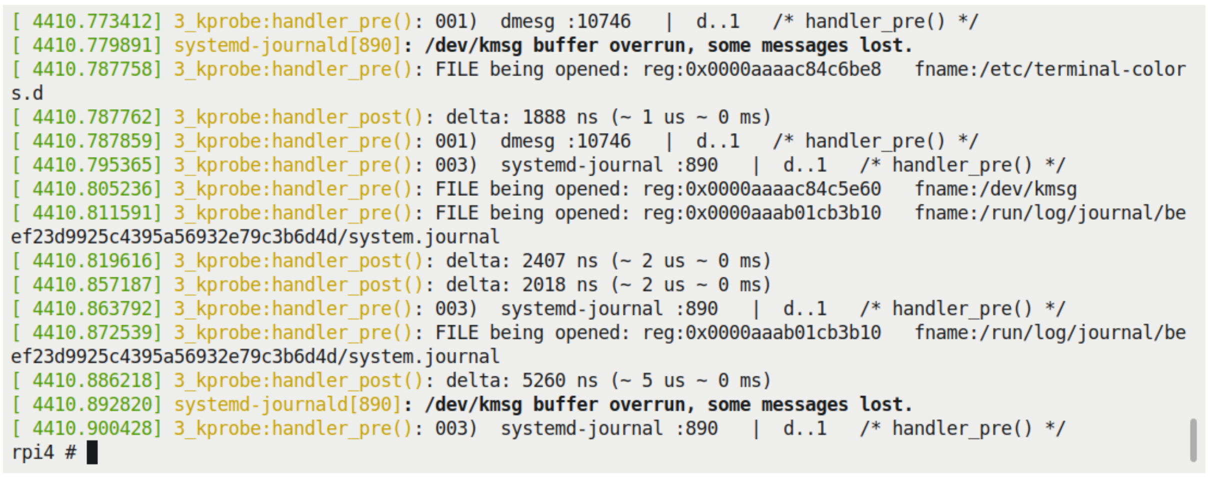
```
[81970.141752] 1_kprobe:handler_pre(): 003)  systemd-journal :395   |   ...1   /*
handler_pre() */
[81970.142245] 1_kprobe:handler_post(): 003)  systemd-journal :395
|  ...1   /* handler_post() */
[81970.143010] 1_kprobe:handler_post(): delta: 100 ns (~ 0 us ~ 0 ms)
```



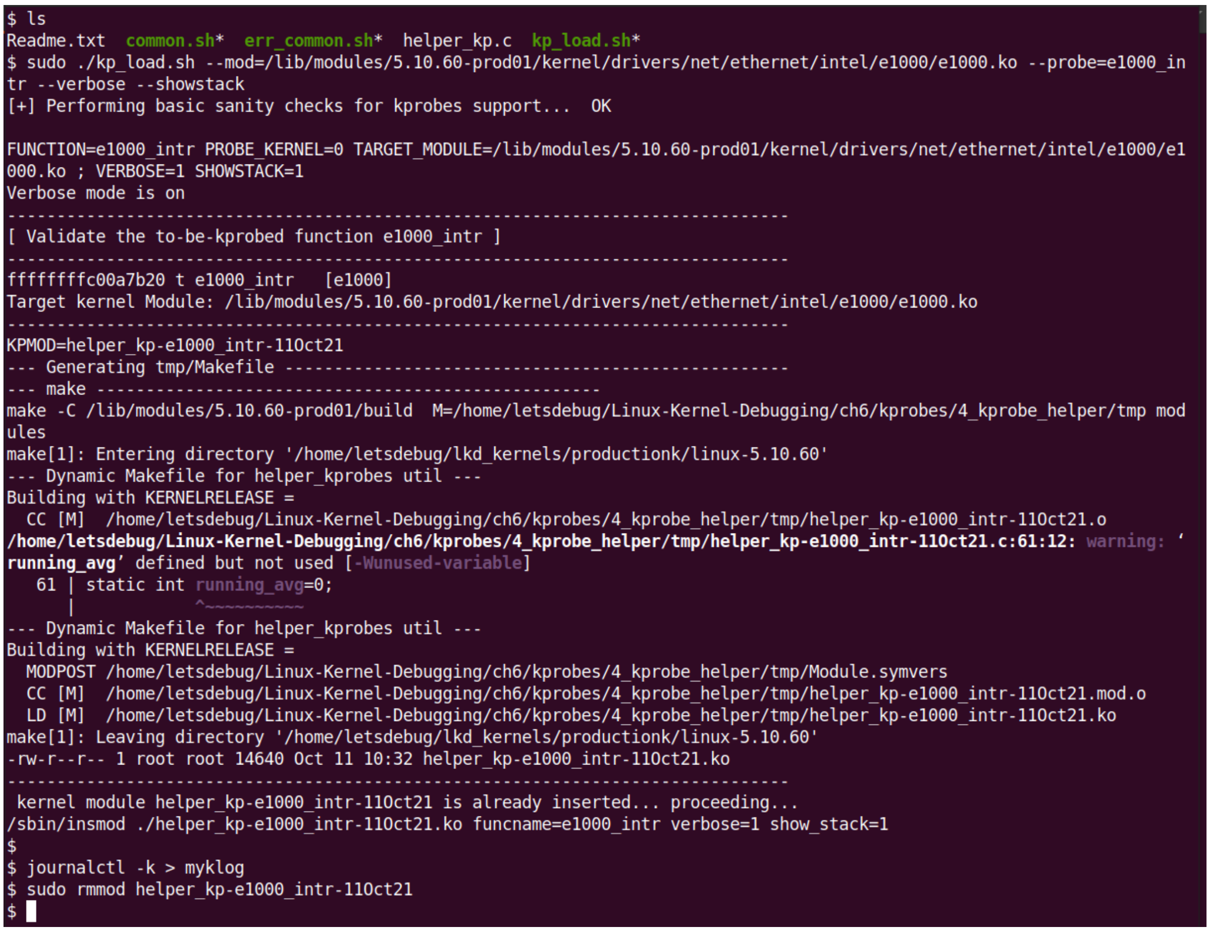
```
Kernel state:
   _-----=> irqs-off          [d]
  / _----=> need-resched      [N]
 | / _---=> hardirq/softirq   [H|h|s]
 || / _--=> preempt-depth     [#]
 ||| /
 ||||
 ||||
 1234   <--- col #

1st col: . = irqs on, d = irqs off
2nd col: . = no pending schedule,
         N = schedule() to be called ASAP
3rd col: . = running in process context
         H|h|s = running in interrupt context
         'h' = hard irq is running
         'H' = hard irq occurred inside a softirq
         ‘s’ = softirq
4th col: preempt-depth (0=>no locks,
         >0=>that many spinlocks being held)
```



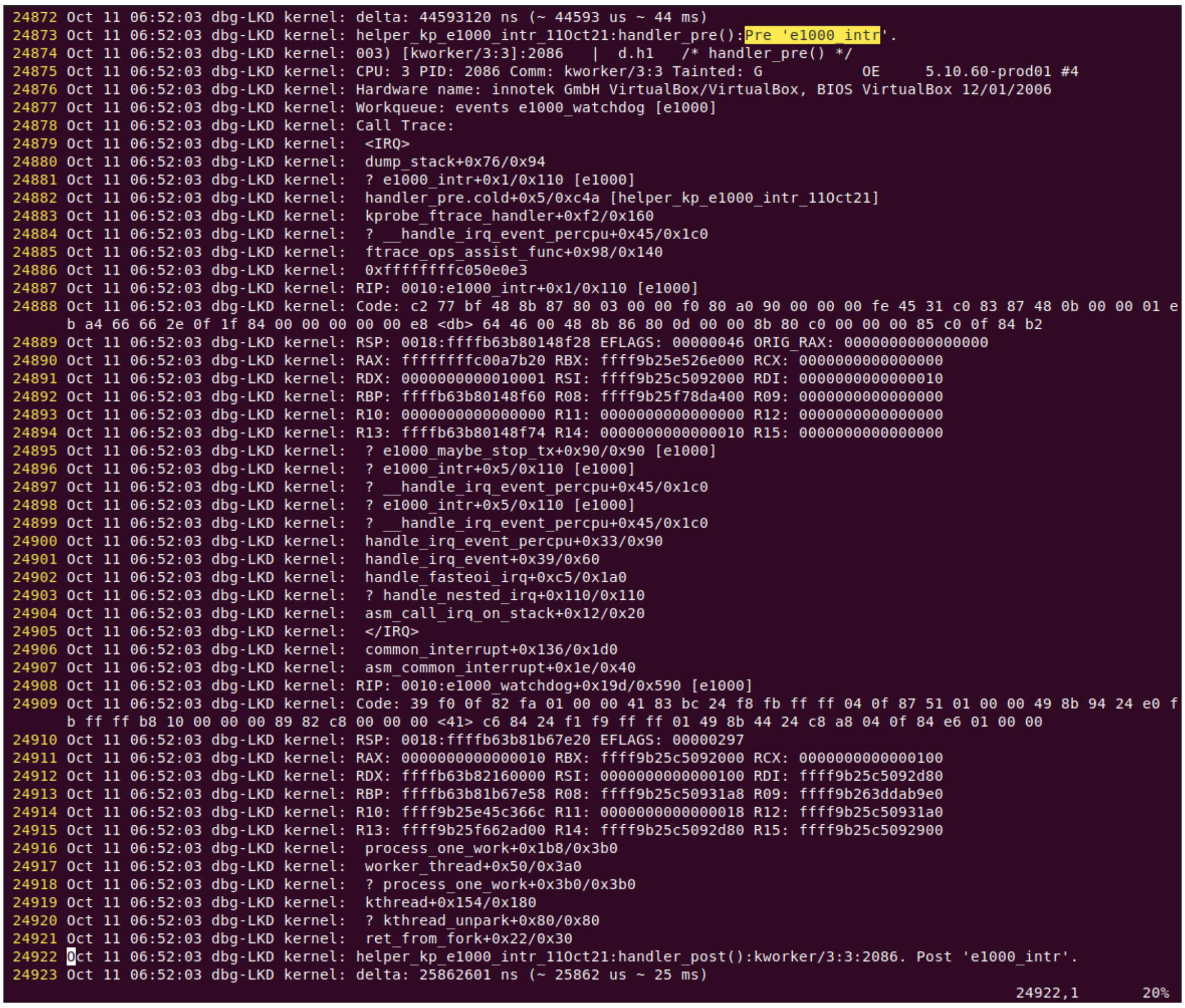
```
[138698.587054] 3_kprobe:handler_pre(): 003)  vi :20612   |  ...1   /* handler_pre() */
[138698.588181] 3_kprobe:handler_pre(): FILE being opened: reg:0x000061bfeaedda10   fname:/etc/vim/after/syntax/sh/
[138698.590315] 3_kprobe:handler_post(): delta: 190 ns (~ 0 us ~ 0 ms)
[138698.591400] 3_kprobe:handler_pre(): 003)  vi :20612   |  ...1   /* handler_pre() */
[138698.592480] 3_kprobe:handler_pre(): FILE being opened: reg:0x000061bfeaedda10   fname:/var/lib/vim/addons/after/syntax/sh/
[138698.594687] 3_kprobe:handler_post(): delta: 190 ns (~ 0 us ~ 0 ms)
[138698.595773] 3_kprobe:handler_pre(): 003)  vi :20612   |  ...1   /* handler_pre() */
[138698.596914] 3_kprobe:handler_pre(): FILE being opened: reg:0x000061bfeaefbc80   fname:/home/letsdebug/.vim/after/syntax/sh/
[138698.599127] 3_kprobe:handler_post(): delta: 176 ns (~ 0 us ~ 0 ms)
[138700.289318] 3_kprobe:handler_pre(): 003)  vi :20612   |  ...1   /* handler_pre() */
[138700.292977] 3_kprobe:handler_pre(): FILE being opened: reg:0x000061bfeaed7980   fname:/home/letsdebug/.viminfo
[138700.300213] 3_kprobe:handler_post(): delta: 855 ns (~ 0 us ~ 0 ms)
[138700.303410] 3_kprobe:handler_pre(): 003)  vi :20612   |  ...1   /* handler_pre() */
[138700.306711] 3_kprobe:handler_pre(): FILE being opened: reg:0x000061bfeaf06640   fname:/home/letsdebug/.viminfo.tmp
[138700.313252] 3_kprobe:handler_post(): delta: 552 ns (~ 0 us ~ 0 ms)
[138700.374248] 3_kprobe:kprobe_lkm_exit(): bye, unregistering kernel probe @ 'do_sys_open'
```



```
[ 4410.773412] 3_kprobe:handler_pre(): 001)  dmesg :10746   |  d..1   /* handler_pre() */
[ 4410.779891] systemd-journald[890]: /dev/kmsg buffer overrun, some messages lost.
[ 4410.787758] 3_kprobe:handler_pre(): FILE being opened: reg:0x0000aaaac84c6be8   fname:/etc/terminal-color
s.d
[ 4410.787762] 3_kprobe:handler_post(): delta: 1888 ns (~ 1 us ~ 0 ms)
[ 4410.787859] 3_kprobe:handler_pre(): 001)  dmesg :10746   |  d..1   /* handler_pre() */
[ 4410.795365] 3_kprobe:handler_pre(): 003)  systemd-journal :890   |  d..1   /* handler_pre() */
[ 4410.805236] 3_kprobe:handler_pre(): FILE being opened: reg:0x0000aaaac84c5e60   fname:/dev/kmsg
[ 4410.811591] 3_kprobe:handler_pre(): FILE being opened: reg:0x0000aaab01cb3b10   fname:/run/log/journal/be
ef23d9925c4395a56932e79c3b6d4d/system.journal
[ 4410.819616] 3_kprobe:handler_post(): delta: 2407 ns (~ 2 us ~ 0 ms)
[ 4410.857187] 3_kprobe:handler_post(): delta: 2018 ns (~ 2 us ~ 0 ms)
[ 4410.863792] 3_kprobe:handler_pre(): 003)  systemd-journal :890   |  d..1   /* handler_pre() */
[ 4410.872539] 3_kprobe:handler_pre(): FILE being opened: reg:0x0000aaab01cb3b10   fname:/run/log/journal/be
ef23d9925c4395a56932e79c3b6d4d/system.journal
[ 4410.886218] 3_kprobe:handler_post(): delta: 5260 ns (~ 5 us ~ 0 ms)
[ 4410.892820] systemd-journald[890]: /dev/kmsg buffer overrun, some messages lost.
[ 4410.900428] 3_kprobe:handler_pre(): 003)  systemd-journal :890   |  d..1   /* handler_pre() */
rpi4 #
```

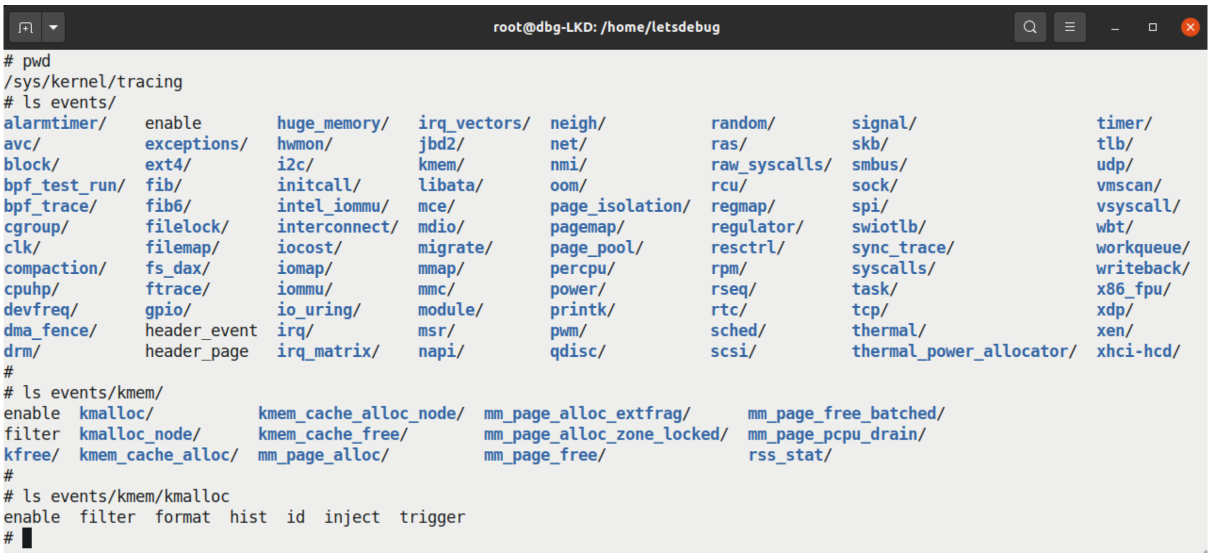


```
$ ls
Readme.txt  common.sh*  err_common.sh*  helper_kp.c  kp_load.sh*
$ sudo ./kp_load.sh --mod=/lib/modules/5.10.60-prod01/kernel/drivers/net/ethernet/intel/e1000/e1000.ko --probe=e1000_in
tr --verbose --showstack
[+] Performing basic sanity checks for kprobes support...  OK

FUNCTION=e1000_intr PROBE_KERNEL=0 TARGET_MODULE=/lib/modules/5.10.60-prod01/kernel/drivers/net/ethernet/intel/e1000/e1
000.ko ; VERBOSE=1 SHOWSTACK=1
Verbose mode is on
--------------------------------------------------------------------------------
[ Validate the to-be-kprobed function e1000_intr ]
--------------------------------------------------------------------------------
ffffffffc00a7b20 t e1000_intr   [e1000]
Target kernel Module: /lib/modules/5.10.60-prod01/kernel/drivers/net/ethernet/intel/e1000/e1000.ko
--------------------------------------------------------------------------------
KPMOD=helper_kp-e1000_intr-11Oct21
--- Generating tmp/Makefile ---------------------------------------------------
--- make --------------------------------------------------
make -C /lib/modules/5.10.60-prod01/build  M=/home/letsdebug/Linux-Kernel-Debugging/ch6/kprobes/4_kprobe_helper/tmp mod
ules
make[1]: Entering directory '/home/letsdebug/lkd_kernels/productionk/linux-5.10.60'
--- Dynamic Makefile for helper_kprobes util ---
Building with KERNELRELEASE =
  CC [M]  /home/letsdebug/Linux-Kernel-Debugging/ch6/kprobes/4_kprobe_helper/tmp/helper_kp-e1000_intr-11Oct21.o
/home/letsdebug/Linux-Kernel-Debugging/ch6/kprobes/4_kprobe_helper/tmp/helper_kp-e1000_intr-11Oct21.c:61:12: warning: ‘
running_avg’ defined but not used [-Wunused-variable]
   61 | static int running_avg=0;
      |            ^~~~~~~~~~~
--- Dynamic Makefile for helper_kprobes util ---
Building with KERNELRELEASE =
  MODPOST /home/letsdebug/Linux-Kernel-Debugging/ch6/kprobes/4_kprobe_helper/tmp/Module.symvers
  CC [M]  /home/letsdebug/Linux-Kernel-Debugging/ch6/kprobes/4_kprobe_helper/tmp/helper_kp-e1000_intr-11Oct21.mod.o
  LD [M]  /home/letsdebug/Linux-Kernel-Debugging/ch6/kprobes/4_kprobe_helper/tmp/helper_kp-e1000_intr-11Oct21.ko
make[1]: Leaving directory '/home/letsdebug/lkd_kernels/productionk/linux-5.10.60'
-rw-r--r-- 1 root root 14640 Oct 11 10:32 helper_kp-e1000_intr-11Oct21.ko
--------------------------------------------------------------------------------
 kernel module helper_kp-e1000_intr-11Oct21 is already inserted... proceeding...
/sbin/insmod ./helper_kp-e1000_intr-11Oct21.ko funcname=e1000_intr verbose=1 show_stack=1
$
$ journalctl -k > myklog
$ sudo rmmod helper_kp-e1000_intr-11Oct21
$ 
```



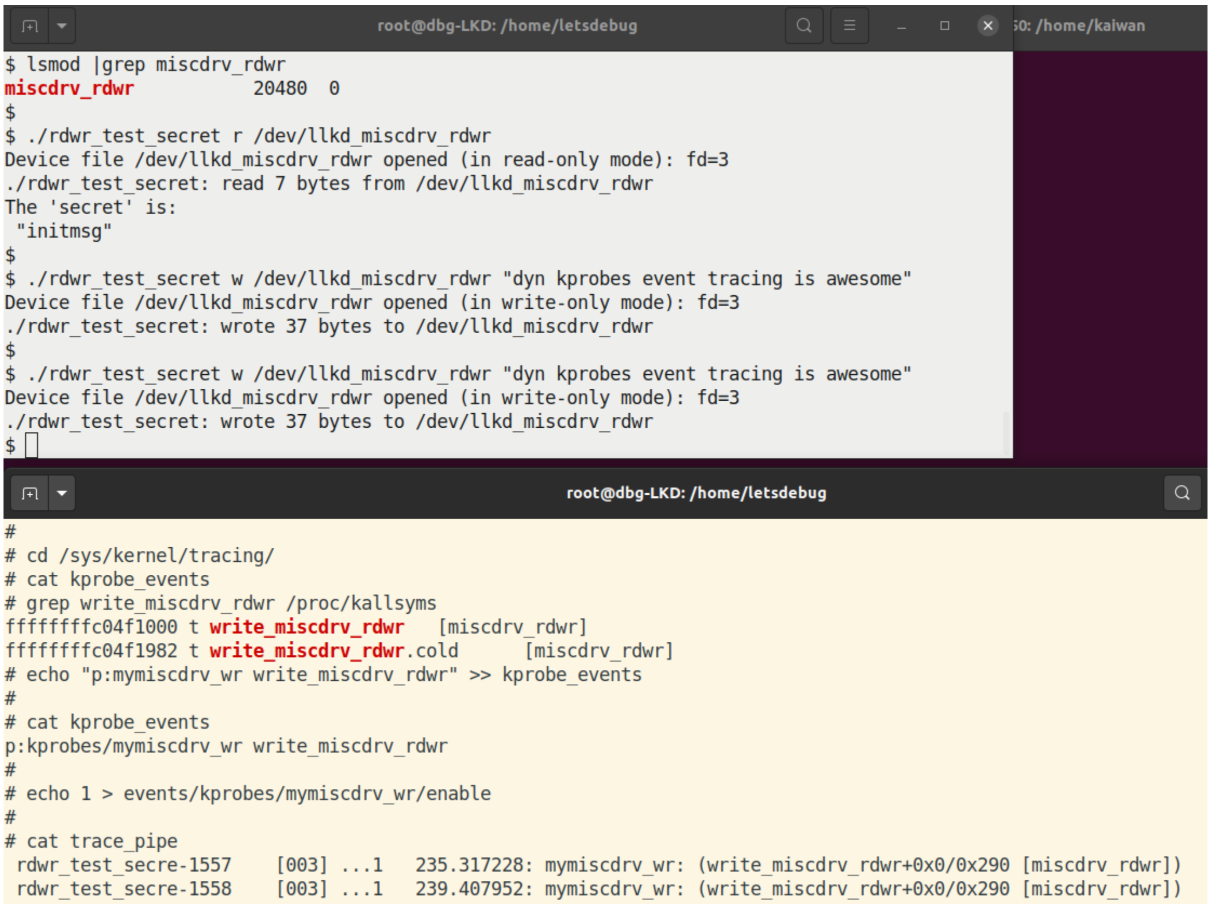
```
Oct 11 06:52:03 dbg-LKD kernel: delta: 44593120 ns (~ 44593 us ~ 44 ms)
Oct 11 06:52:03 dbg-LKD kernel: helper_kp_e1000_intr_11Oct21:handler_pre():Pre 'e1000_intr'.
Oct 11 06:52:03 dbg-LKD kernel: 003) [kworker/3:3]:2086   |  d.h1   /* handler_pre() */
Oct 11 06:52:03 dbg-LKD kernel: CPU: 3 PID: 2086 Comm: kworker/3:3 Tainted: G           OE     5.10.60-prod01 #4
Oct 11 06:52:03 dbg-LKD kernel: Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
Oct 11 06:52:03 dbg-LKD kernel: Workqueue: events e1000_watchdog [e1000]
Oct 11 06:52:03 dbg-LKD kernel: Call Trace:
Oct 11 06:52:03 dbg-LKD kernel:  <IRQ>
Oct 11 06:52:03 dbg-LKD kernel:  dump_stack+0x76/0x94
Oct 11 06:52:03 dbg-LKD kernel:  ? e1000_intr+0x1/0x110 [e1000]
Oct 11 06:52:03 dbg-LKD kernel:  handler_pre.cold+0x5/0xc4a [helper_kp_e1000_intr_11Oct21]
Oct 11 06:52:03 dbg-LKD kernel:  kprobe_ftrace_handler+0xf2/0x160
Oct 11 06:52:03 dbg-LKD kernel:  ? __handle_irq_event_percpu+0x45/0x1c0
Oct 11 06:52:03 dbg-LKD kernel:  ftrace_ops_assist_func+0x98/0x140
Oct 11 06:52:03 dbg-LKD kernel:  0xffffffffc050e0e3
Oct 11 06:52:03 dbg-LKD kernel: RIP: 0010:e1000_intr+0x1/0x110 [e1000]
Oct 11 06:52:03 dbg-LKD kernel: Code: c2 77 bf 48 8b 87 80 03 00 00 f0 80 a0 90 00 00 00 fe 45 31 c0 83 87 48 0b 00 00 01 e
b a4 66 66 2e 0f 1f 84 00 00 00 00 00 e8 <db> 64 46 00 48 8b 86 80 0d 00 00 8b 80 c0 00 00 00 85 c0 0f 84 b2
Oct 11 06:52:03 dbg-LKD kernel: RSP: 0018:ffffb63b80148f28 EFLAGS: 00000046 ORIG_RAX: 0000000000000000
Oct 11 06:52:03 dbg-LKD kernel: RAX: ffffffffc00a7b20 RBX: ffff9b25e526e000 RCX: 0000000000000000
Oct 11 06:52:03 dbg-LKD kernel: RDX: 0000000000010001 RSI: ffff9b25c5092000 RDI: 0000000000000010
Oct 11 06:52:03 dbg-LKD kernel: RBP: ffffb63b80148f60 R08: ffff9b25f78da400 R09: 0000000000000000
Oct 11 06:52:03 dbg-LKD kernel: R10: 0000000000000000 R11: 0000000000000000 R12: 0000000000000000
Oct 11 06:52:03 dbg-LKD kernel: R13: ffffb63b80148f74 R14: 0000000000000010 R15: 0000000000000000
Oct 11 06:52:03 dbg-LKD kernel:  ? e1000_maybe_stop_tx+0x90/0x90 [e1000]
Oct 11 06:52:03 dbg-LKD kernel:  ? e1000_intr+0x5/0x110 [e1000]
Oct 11 06:52:03 dbg-LKD kernel:  ? __handle_irq_event_percpu+0x45/0x1c0
Oct 11 06:52:03 dbg-LKD kernel:  ? e1000_intr+0x5/0x110 [e1000]
Oct 11 06:52:03 dbg-LKD kernel:  ? __handle_irq_event_percpu+0x45/0x1c0
Oct 11 06:52:03 dbg-LKD kernel:  handle_irq_event_percpu+0x33/0x90
Oct 11 06:52:03 dbg-LKD kernel:  handle_irq_event+0x39/0x60
Oct 11 06:52:03 dbg-LKD kernel:  handle_fasteoi_irq+0xc5/0x1a0
Oct 11 06:52:03 dbg-LKD kernel:  ? handle_nested_irq+0x110/0x110
Oct 11 06:52:03 dbg-LKD kernel:  asm_call_irq_on_stack+0x12/0x20
Oct 11 06:52:03 dbg-LKD kernel:  </IRQ>
Oct 11 06:52:03 dbg-LKD kernel:  common_interrupt+0x136/0x1d0
Oct 11 06:52:03 dbg-LKD kernel:  asm_common_interrupt+0x1e/0x40
Oct 11 06:52:03 dbg-LKD kernel: RIP: 0010:e1000_watchdog+0x19d/0x590 [e1000]
Oct 11 06:52:03 dbg-LKD kernel: Code: 39 f0 0f 82 fa 01 00 00 41 83 bc 24 f8 fb ff ff 04 0f 87 51 01 00 00 49 8b 94 24 e0 f
b ff ff b8 10 00 00 00 89 82 c8 00 00 00 <41> c6 84 24 f1 f9 ff ff 01 49 8b 44 24 c8 a8 04 0f 84 e6 01 00 00
Oct 11 06:52:03 dbg-LKD kernel: RSP: 0018:ffffb63b81b67e20 EFLAGS: 00000297
Oct 11 06:52:03 dbg-LKD kernel: RAX: 0000000000000010 RBX: ffff9b25c5092000 RCX: 0000000000000100
Oct 11 06:52:03 dbg-LKD kernel: RDX: ffffb63b82160000 RSI: 0000000000000100 RDI: ffff9b25c5092d80
Oct 11 06:52:03 dbg-LKD kernel: RBP: ffffb63b81b67e58 R08: ffff9b25c50931a8 R09: ffff9b263ddab9e0
Oct 11 06:52:03 dbg-LKD kernel: R10: ffff9b25e45c366c R11: 0000000000000018 R12: ffff9b25c50931a0
Oct 11 06:52:03 dbg-LKD kernel: R13: ffff9b25f662ad00 R14: ffff9b25c5092d80 R15: ffff9b25c5092900
Oct 11 06:52:03 dbg-LKD kernel:  process_one_work+0x1b8/0x3b0
Oct 11 06:52:03 dbg-LKD kernel:  worker_thread+0x50/0x3a0
Oct 11 06:52:03 dbg-LKD kernel:  ? process_one_work+0x3b0/0x3b0
Oct 11 06:52:03 dbg-LKD kernel:  kthread+0x154/0x180
Oct 11 06:52:03 dbg-LKD kernel:  ? kthread_unpark+0x80/0x80
Oct 11 06:52:03 dbg-LKD kernel:  ret_from_fork+0x22/0x30
Oct 11 06:52:03 dbg-LKD kernel: helper_kp_e1000_intr_11Oct21:handler_post():kworker/3:3:2086. Post 'e1000_intr'.
Oct 11 06:52:03 dbg-LKD kernel: delta: 25862601 ns (~ 25862 us ~ 25 ms)
                                                                                              24922,1       20%
```

```
# kprobe-perf
USAGE: kprobe [-FhHsv] [-d secs] [-p PID] [-L TID] kprobe_definition [filter]
                 -F              # force. trace despite warnings.
                 -d seconds      # trace duration, and use buffers
                 -p PID          # PID to match on events
                 -L TID          # thread id to match on events
                 -v              # view format file (don't trace)
                 -H              # include column headers
                 -s              # show kernel stack traces
                 -h              # this usage message

Note that these examples may need modification to match your kernel
version's function names and platform's register usage.
   eg,
       kprobe p:do_sys_open
                                 # trace open() entry
       kprobe r:do_sys_open
                                 # trace open() return
       kprobe 'r:do_sys_open $retval'
                                 # trace open() return value
       kprobe 'r:myopen do_sys_open $retval'
                                 # use a custom probe name
       kprobe 'p:myopen do_sys_open mode=%cx:u16'
                                 # trace open() file mode
       kprobe 'p:myopen do_sys_open filename=+0(%si):string'
                                 # trace open() with filename
       kprobe -s 'p:myprobe tcp_retransmit_skb'
                                 # show kernel stacks
       kprobe 'p:do_sys_open file=+0(%si):string' 'file ~ "*stat"'
                                 # opened files ending in "stat"

See the man page and example file for more info.
#
```



```
# pwd
/sys/kernel/tracing
# ls events/
alarmtimer/    enable        huge_memory/   irq_vectors/  neigh/           random/        signal/                   timer/
avc/           exceptions/   hwmon/         jbd2/         net/             ras/           skb/                      tlb/
block/         ext4/         i2c/           kmem/         nmi/             raw_syscalls/  smbus/                    udp/
bpf_test_run/  fib/          initcall/      libata/       oom/             rcu/           sock/                     vmscan/
bpf_trace/     fib6/         intel_iommu/   mce/          page_isolation/  regmap/        spi/                      vsyscall/
cgroup/        filelock/     interconnect/  mdio/         pagemap/         regulator/     swiotlb/                  wbt/
clk/           filemap/      iocost/        migrate/      page_pool/       resctrl/       sync_trace/               workqueue/
compaction/    fs_dax/       iomap/         mmap/         percpu/          rpm/           syscalls/                 writeback/
cpuhp/         ftrace/       iommu/         mmc/          power/           rseq/          task/                     x86_fpu/
devfreq/       gpio/         io_uring/      module/       printk/          rtc/           tcp/                      xdp/
dma_fence/     header_event  irq/           msr/          pwm/             sched/         thermal/                  xen/
drm/           header_page   irq_matrix/    napi/         qdisc/           scsi/          thermal_power_allocator/  xhci-hcd/
#
# ls events/kmem/
enable  kmalloc/           kmem_cache_alloc_node/  mm_page_alloc_extfrag/      mm_page_free_batched/
filter  kmalloc_node/      kmem_cache_free/        mm_page_alloc_zone_locked/  mm_page_pcpu_drain/
kfree/  kmem_cache_alloc/  mm_page_alloc/          mm_page_free/               rss_stat/
#
# ls events/kmem/kmalloc
enable  filter  format  hist  id  inject  trigger
# 
```

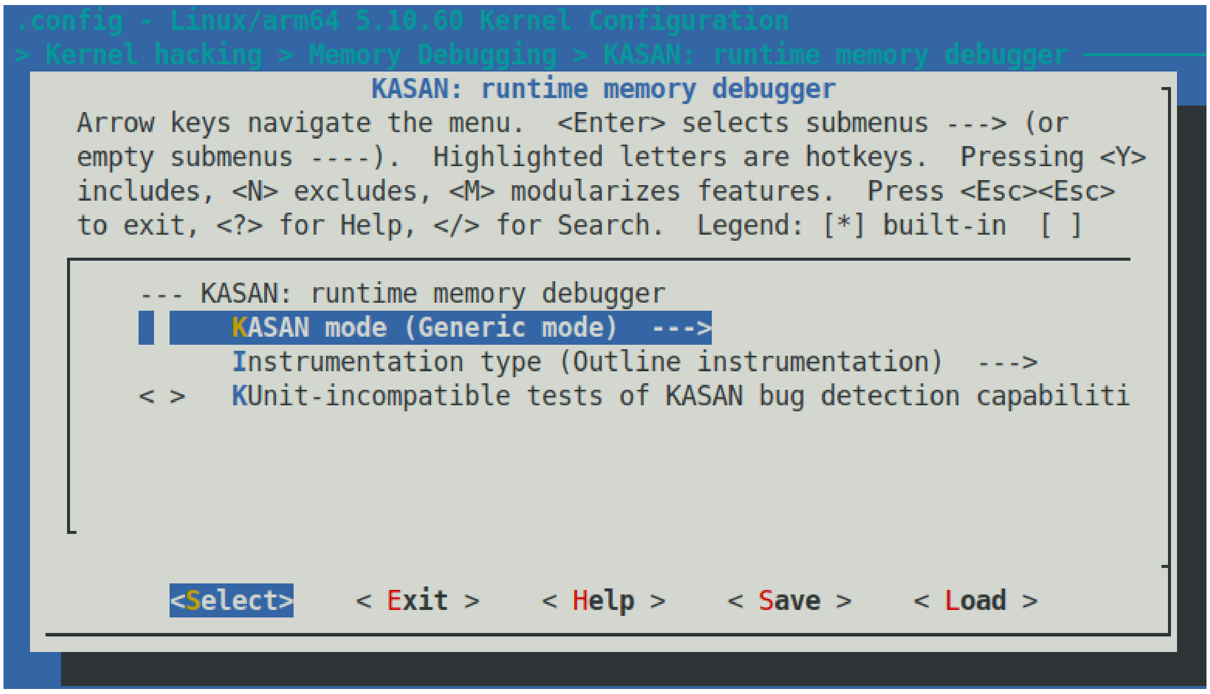
```
rpi # pwd
/sys/kernel/debug/tracing
rpi # cat events/kmem/kmalloc/enable
0
rpi # echo 1 > events/kmem/kmalloc/enable
rpi #
rpi # cat trace_pipe
            sshd-680     [000] ....   700.723280: kmalloc: call_site=__alloc_skb+0x70/0x164 ptr=236acdbb byte
s_req=576 bytes_alloc=1024 gfp_flags=GFP_KERNEL|__GFP_NOWARN|__GFP_NOMEMALLOC
            sshd-680     [000] ....   700.723391: kmalloc: call_site=pskb_expand_head+0x70/0x33c ptr=f5e025aa
 bytes_req=1024 bytes_alloc=1024 gfp_flags=GFP_ATOMIC|__GFP_NOWARN|__GFP_NOMEMALLOC
    kworker/u2:1-56      [000] ....   700.723674: kmalloc: call_site=__alloc_skb+0x70/0x164 ptr=a25030bf byte
s_req=352 bytes_alloc=512 gfp_flags=GFP_ATOMIC|__GFP_NOWARN|__GFP_NOMEMALLOC
    kworker/u2:1-56      [000] ....   700.725507: kmalloc: call_site=__alloc_skb+0x70/0x164 ptr=a25030bf byte
s_req=352 bytes_alloc=512 gfp_flags=GFP_ATOMIC|__GFP_NOWARN|__GFP_NOMEMALLOC
    kworker/u2:1-56      [000] ....   700.725607: kmalloc: call_site=__alloc_skb+0x70/0x164 ptr=a25030bf byte
s_req=384 bytes_alloc=512 gfp_flags=GFP_ATOMIC|__GFP_NOWARN|__GFP_NOMEMALLOC
```



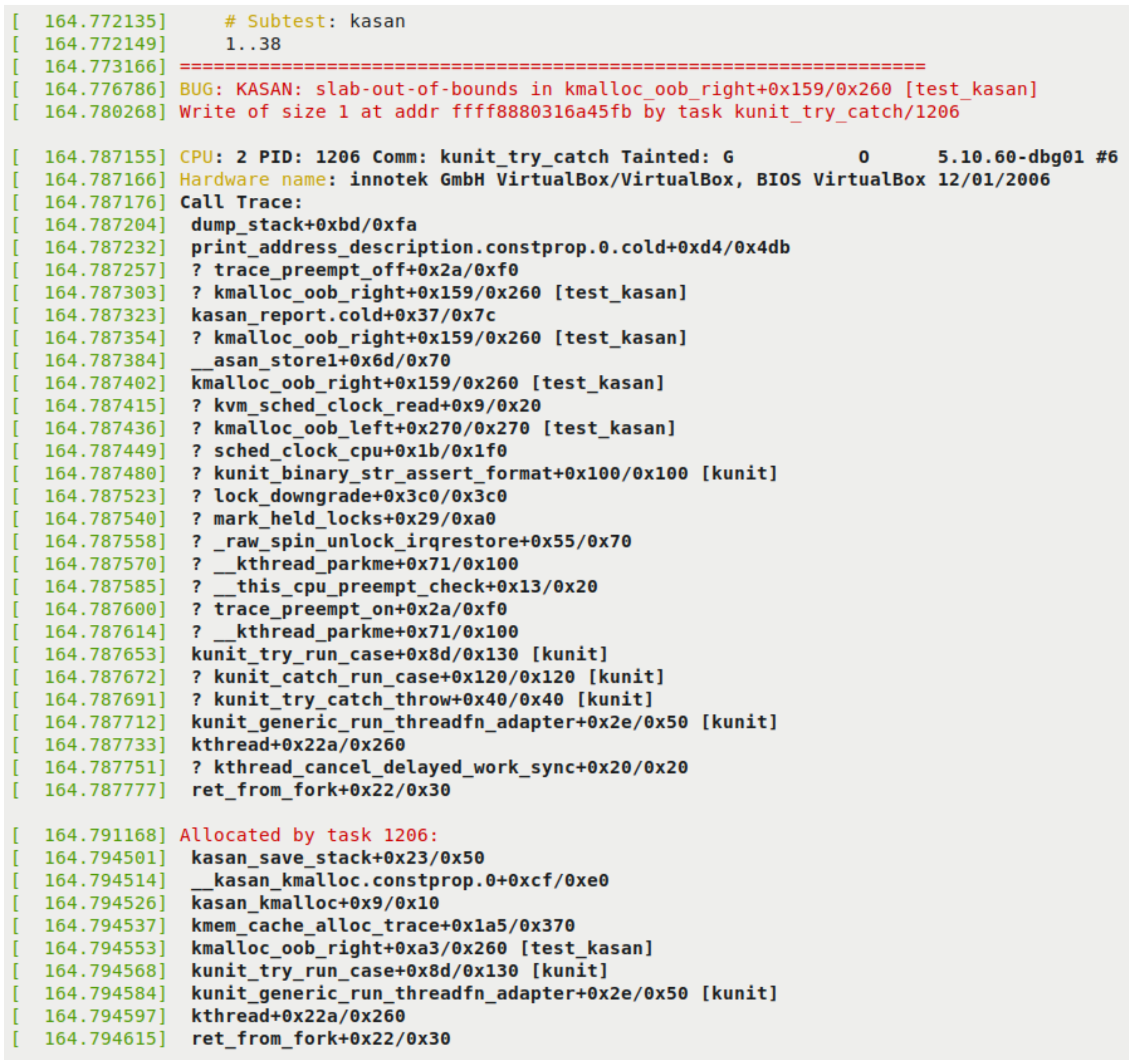
```
$ lsmod |grep miscdrv_rdwr
miscdrv_rdwr           20480  0
$
$ ./rdwr_test_secret r /dev/llkd_miscdrv_rdwr
Device file /dev/llkd_miscdrv_rdwr opened (in read-only mode): fd=3
./rdwr_test_secret: read 7 bytes from /dev/llkd_miscdrv_rdwr
The 'secret' is:
 "initmsg"
$
$ ./rdwr_test_secret w /dev/llkd_miscdrv_rdwr "dyn kprobes event tracing is awesome"
Device file /dev/llkd_miscdrv_rdwr opened (in write-only mode): fd=3
./rdwr_test_secret: wrote 37 bytes to /dev/llkd_miscdrv_rdwr
$
$ ./rdwr_test_secret w /dev/llkd_miscdrv_rdwr "dyn kprobes event tracing is awesome"
Device file /dev/llkd_miscdrv_rdwr opened (in write-only mode): fd=3
./rdwr_test_secret: wrote 37 bytes to /dev/llkd_miscdrv_rdwr
$
```

```
#
# cd /sys/kernel/tracing/
# cat kprobe_events
# grep write_miscdrv_rdwr /proc/kallsyms
ffffffffc04f1000 t write_miscdrv_rdwr   [miscdrv_rdwr]
ffffffffc04f1982 t write_miscdrv_rdwr.cold      [miscdrv_rdwr]
# echo "p:mymiscdrv_wr write_miscdrv_rdwr" >> kprobe_events
#
# cat kprobe_events
p:kprobes/mymiscdrv_wr write_miscdrv_rdwr
#
# echo 1 > events/kprobes/mymiscdrv_wr/enable
#
# cat trace_pipe
 rdwr_test_secre-1557    [003] ...1   235.317228: mymiscdrv_wr: (write_miscdrv_rdwr+0x0/0x290 [miscdrv_rdwr])
 rdwr_test_secre-1558    [003] ...1   239.407952: mymiscdrv_wr: (write_miscdrv_rdwr+0x0/0x290 [miscdrv_rdwr])
```

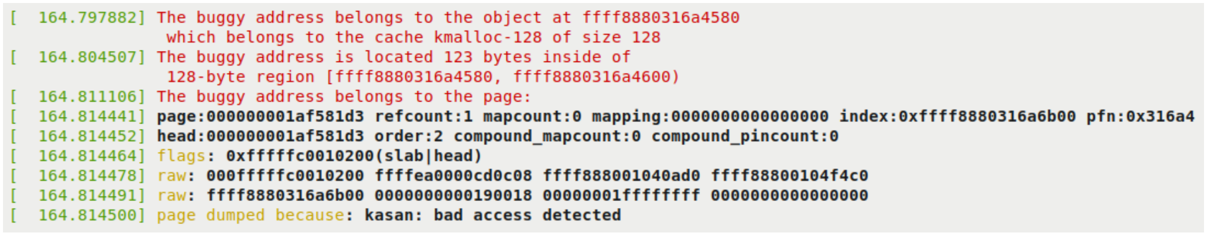
```
Generically:
foo()   ----------->       sys_foo() → do_foo()


getpgid() --------> sys_getpgid() → do_getpgid()

open() ----------->     sys_open() → do_sys_open() → do_sys_openat2() → do_filp_open() → …

execve() --------->   sys_execve() → do_execve()

User mode                                      Kernel mode
```



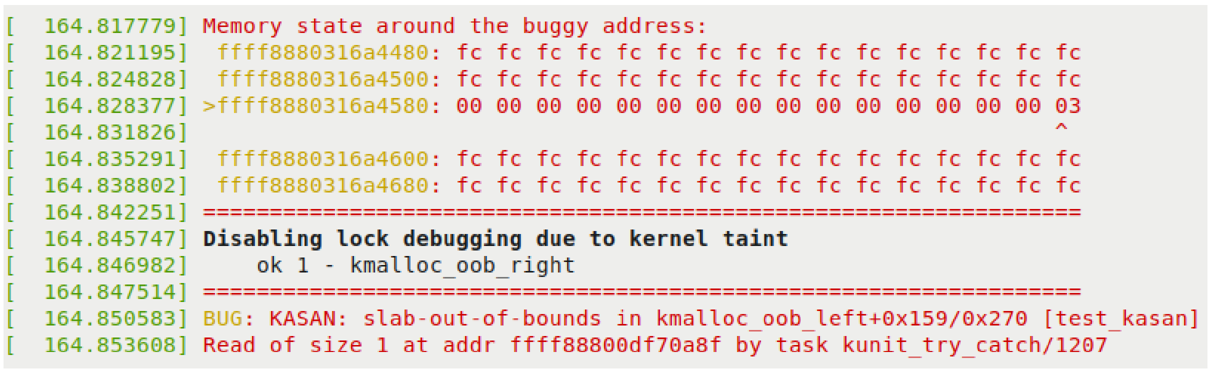
```
.config - Linux/arm64 5.10.60 Kernel Configuration
> Kernel hacking > Memory Debugging > KASAN: runtime memory debugger ─────
                      KASAN: runtime memory debugger
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or
  empty submenus ----).  Highlighted letters are hotkeys.  Pressing <Y>
  includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc>
  to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ]

      --- KASAN: runtime memory debugger
            KASAN mode (Generic mode)  --->
            Instrumentation type (Outline instrumentation)  --->
      < >   KUnit-incompatible tests of KASAN bug detection capabiliti

         <Select>    < Exit >    < Help >    < Save >    < Load >
```



```
[  164.772135]     # Subtest: kasan
[  164.772149]     1..38
[  164.773166] ==================================================================
[  164.776786] BUG: KASAN: slab-out-of-bounds in kmalloc_oob_right+0x159/0x260 [test_kasan]
[  164.780268] Write of size 1 at addr ffff8880316a45fb by task kunit_try_catch/1206

[  164.787155] CPU: 2 PID: 1206 Comm: kunit_try_catch Tainted: G           O      5.10.60-dbg01 #6
[  164.787166] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[  164.787176] Call Trace:
[  164.787204]  dump_stack+0xbd/0xfa
[  164.787232]  print_address_description.constprop.0.cold+0xd4/0x4db
[  164.787257]  ? trace_preempt_off+0x2a/0xf0
[  164.787303]  ? kmalloc_oob_right+0x159/0x260 [test_kasan]
[  164.787323]  kasan_report.cold+0x37/0x7c
[  164.787354]  ? kmalloc_oob_right+0x159/0x260 [test_kasan]
[  164.787384]  __asan_store1+0x6d/0x70
[  164.787402]  kmalloc_oob_right+0x159/0x260 [test_kasan]
[  164.787415]  ? kvm_sched_clock_read+0x9/0x20
[  164.787436]  ? kmalloc_oob_left+0x270/0x270 [test_kasan]
[  164.787449]  ? sched_clock_cpu+0x1b/0x1f0
[  164.787480]  ? kunit_binary_str_assert_format+0x100/0x100 [kunit]
[  164.787523]  ? lock_downgrade+0x3c0/0x3c0
[  164.787540]  ? mark_held_locks+0x29/0xa0
[  164.787558]  ? _raw_spin_unlock_irqrestore+0x55/0x70
[  164.787570]  ? __kthread_parkme+0x71/0x100
[  164.787585]  ? __this_cpu_preempt_check+0x13/0x20
[  164.787600]  ? trace_preempt_on+0x2a/0xf0
[  164.787614]  ? __kthread_parkme+0x71/0x100
[  164.787653]  kunit_try_run_case+0x8d/0x130 [kunit]
[  164.787672]  ? kunit_catch_run_case+0x120/0x120 [kunit]
[  164.787691]  ? kunit_try_catch_throw+0x40/0x40 [kunit]
[  164.787712]  kunit_generic_run_threadfn_adapter+0x2e/0x50 [kunit]
[  164.787733]  kthread+0x22a/0x260
[  164.787751]  ? kthread_cancel_delayed_work_sync+0x20/0x20
[  164.787777]  ret_from_fork+0x22/0x30

[  164.791168] Allocated by task 1206:
[  164.794501]  kasan_save_stack+0x23/0x50
[  164.794514]  __kasan_kmalloc.constprop.0+0xcf/0xe0
[  164.794526]  kasan_kmalloc+0x9/0x10
[  164.794537]  kmem_cache_alloc_trace+0x1a5/0x370
[  164.794553]  kmalloc_oob_right+0xa3/0x260 [test_kasan]
[  164.794568]  kunit_try_run_case+0x8d/0x130 [kunit]
[  164.794584]  kunit_generic_run_threadfn_adapter+0x2e/0x50 [kunit]
[  164.794597]  kthread+0x22a/0x260
[  164.794615]  ret_from_fork+0x22/0x30
```

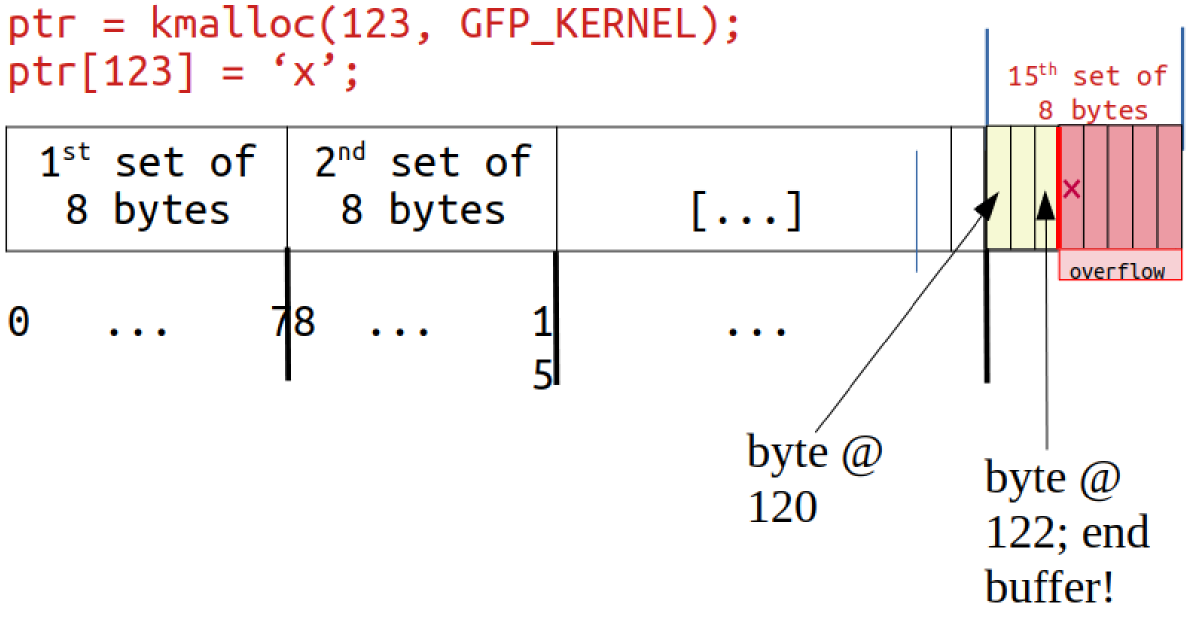


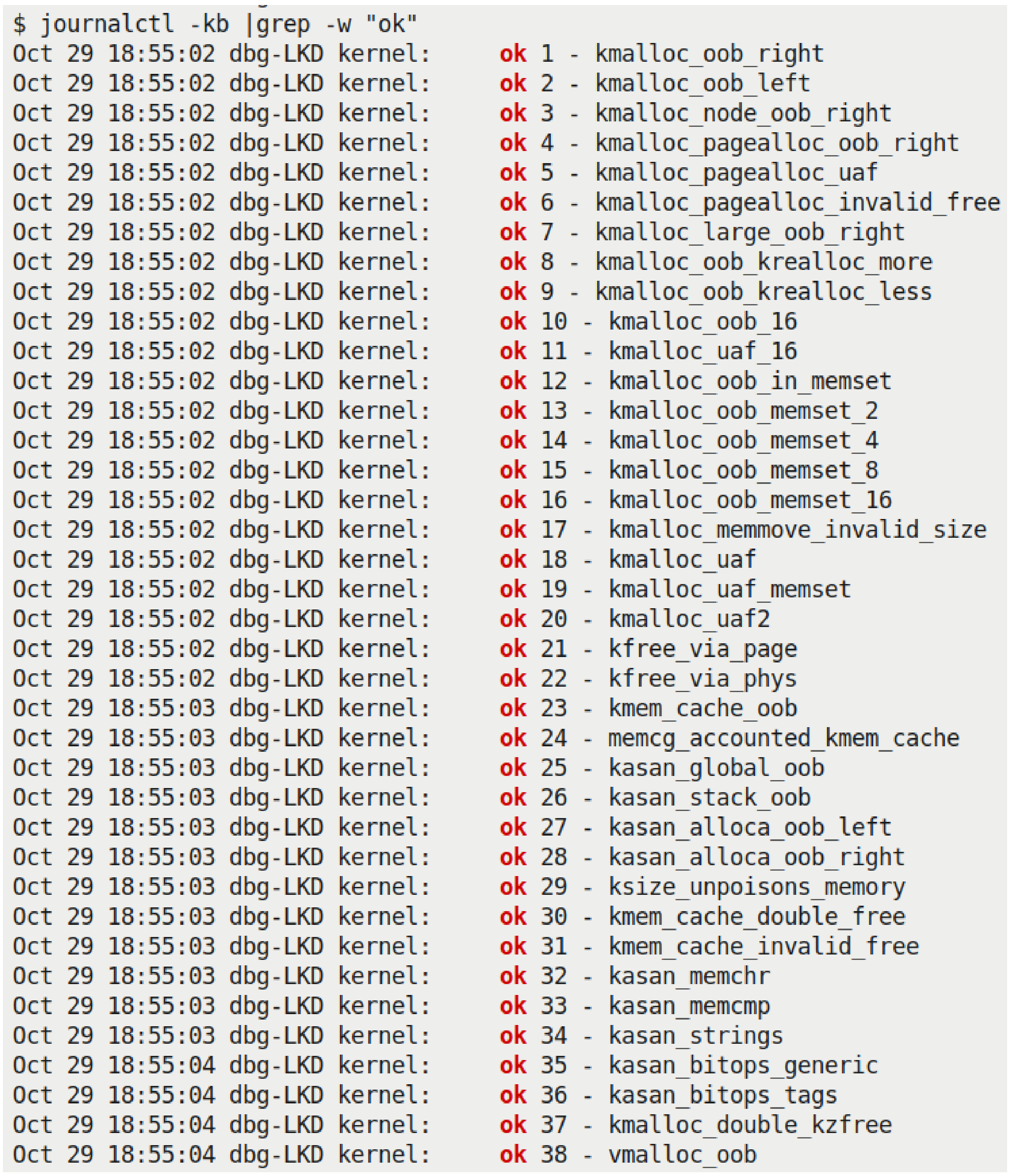
```
[  164.797882] The buggy address belongs to the object at ffff8880316a4580
                which belongs to the cache kmalloc-128 of size 128
[  164.804507] The buggy address is located 123 bytes inside of
                128-byte region [ffff8880316a4580, ffff8880316a4600)
[  164.811106] The buggy address belongs to the page:
[  164.814441] page:000000001af581d3 refcount:1 mapcount:0 mapping:0000000000000000 index:0xffff8880316a6b00 pfn:0x316a4
[  164.814452] head:000000001af581d3 order:2 compound_mapcount:0 compound_pincount:0
[  164.814464] flags: 0xfffffc0010200(slab|head)
[  164.814478] raw: 000fffffc0010200 ffffea0000cd0c08 ffff888001040ad0 ffff88800104f4c0
[  164.814491] raw: ffff8880316a6b00 0000000000190018 00000001ffffffff 0000000000000000
[  164.814500] page dumped because: kasan: bad access detected
```



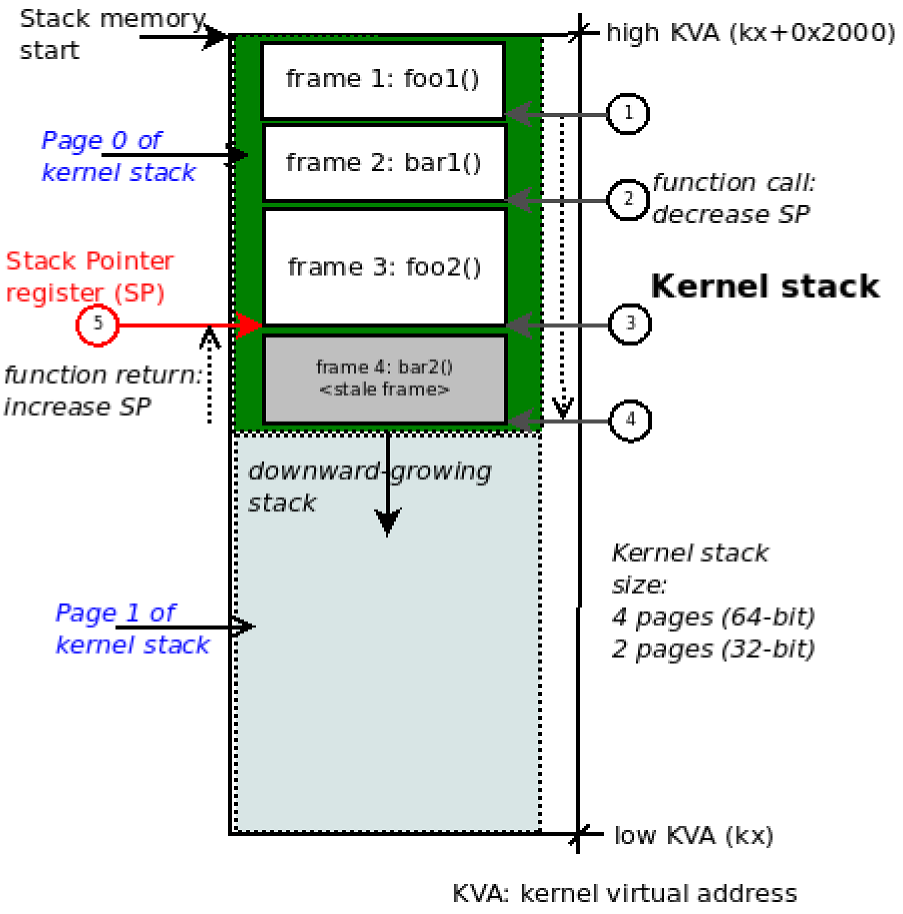
```
[  164.817779] Memory state around the buggy address:
[  164.821195]  ffff8880316a4480: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
[  164.824828]  ffff8880316a4500: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
[  164.828377] >ffff8880316a4580: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03
[  164.831826]                                                                ^
[  164.835291]  ffff8880316a4600: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
[  164.838802]  ffff8880316a4680: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
[  164.842251] ==================================================================
[  164.845747] Disabling lock debugging due to kernel taint
[  164.846982]     ok 1 - kmalloc_oob_right
[  164.847514] ==================================================================
[  164.850583] BUG: KASAN: slab-out-of-bounds in kmalloc_oob_left+0x159/0x270 [test_kasan]
[  164.853608] Read of size 1 at addr ffff88800df70a8f by task kunit_try_catch/1207
```

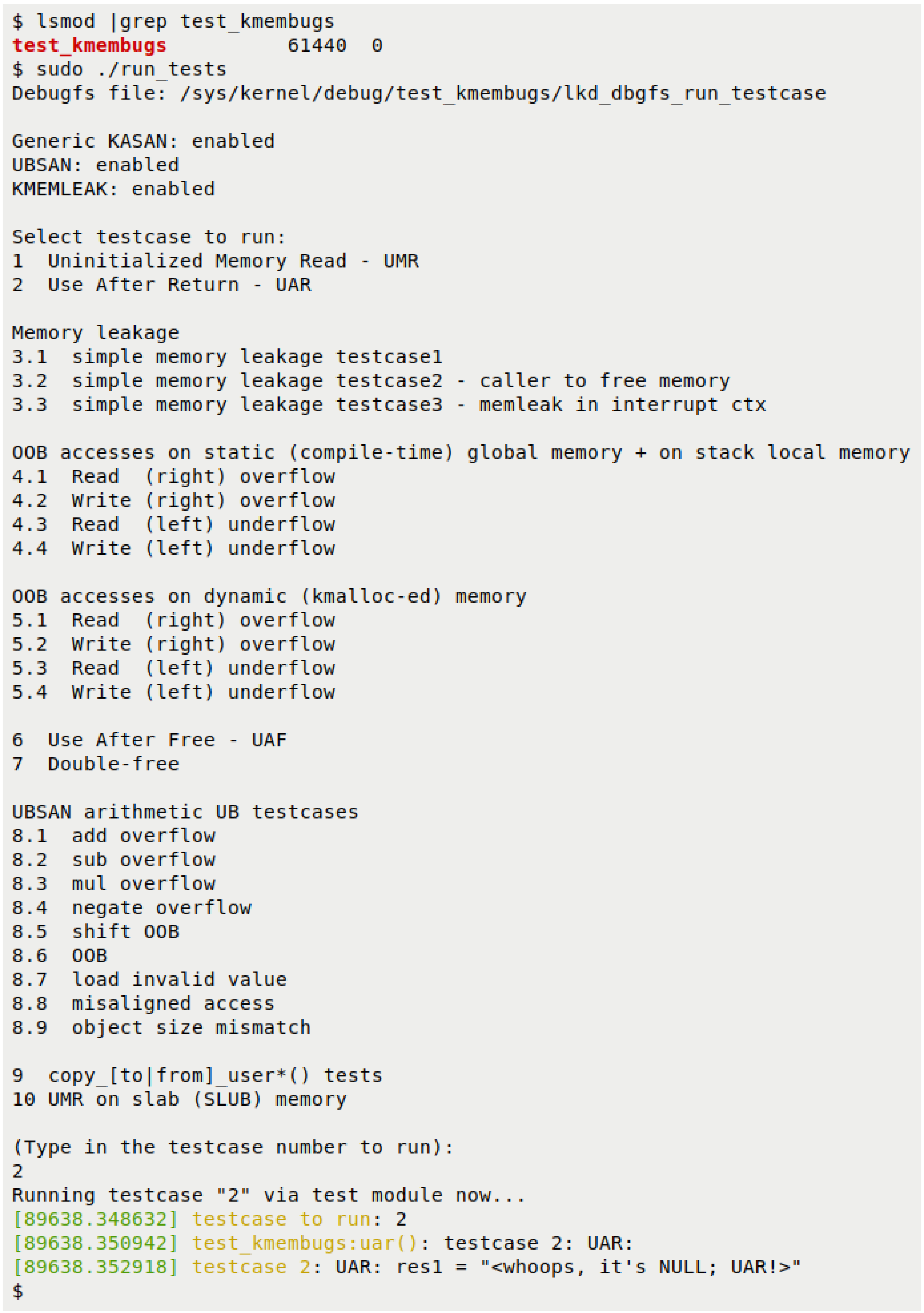
```
ptr = kmalloc(123, GFP_KERNEL);
ptr[123] = ‘x’;
```



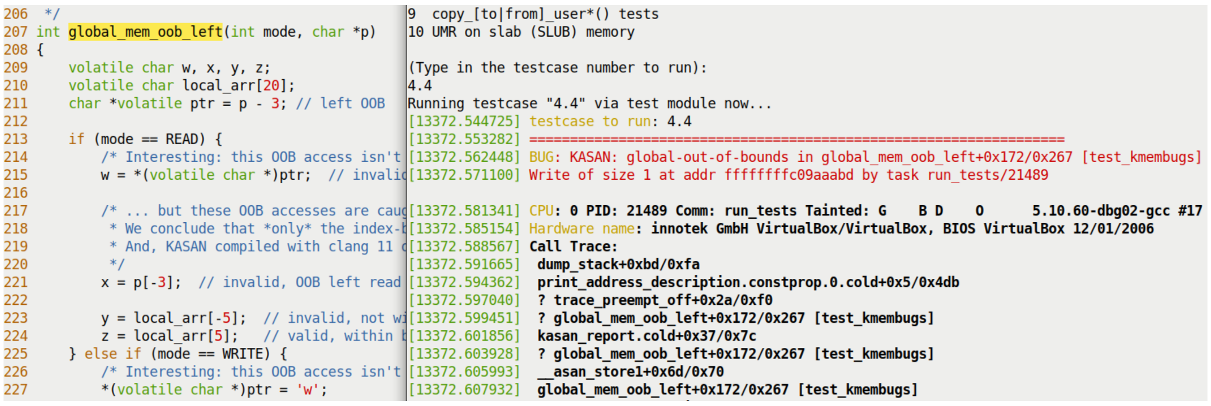


```
$ journalctl -kb |grep -w "ok"
Oct 29 18:55:02 dbg-LKD kernel:     ok 1 - kmalloc_oob_right
Oct 29 18:55:02 dbg-LKD kernel:     ok 2 - kmalloc_oob_left
Oct 29 18:55:02 dbg-LKD kernel:     ok 3 - kmalloc_node_oob_right
Oct 29 18:55:02 dbg-LKD kernel:     ok 4 - kmalloc_pagealloc_oob_right
Oct 29 18:55:02 dbg-LKD kernel:     ok 5 - kmalloc_pagealloc_uaf
Oct 29 18:55:02 dbg-LKD kernel:     ok 6 - kmalloc_pagealloc_invalid_free
Oct 29 18:55:02 dbg-LKD kernel:     ok 7 - kmalloc_large_oob_right
Oct 29 18:55:02 dbg-LKD kernel:     ok 8 - kmalloc_oob_krealloc_more
Oct 29 18:55:02 dbg-LKD kernel:     ok 9 - kmalloc_oob_krealloc_less
Oct 29 18:55:02 dbg-LKD kernel:     ok 10 - kmalloc_oob_16
Oct 29 18:55:02 dbg-LKD kernel:     ok 11 - kmalloc_uaf_16
Oct 29 18:55:02 dbg-LKD kernel:     ok 12 - kmalloc_oob_in_memset
Oct 29 18:55:02 dbg-LKD kernel:     ok 13 - kmalloc_oob_memset_2
Oct 29 18:55:02 dbg-LKD kernel:     ok 14 - kmalloc_oob_memset_4
Oct 29 18:55:02 dbg-LKD kernel:     ok 15 - kmalloc_oob_memset_8
Oct 29 18:55:02 dbg-LKD kernel:     ok 16 - kmalloc_oob_memset_16
Oct 29 18:55:02 dbg-LKD kernel:     ok 17 - kmalloc_memmove_invalid_size
Oct 29 18:55:02 dbg-LKD kernel:     ok 18 - kmalloc_uaf
Oct 29 18:55:02 dbg-LKD kernel:     ok 19 - kmalloc_uaf_memset
Oct 29 18:55:02 dbg-LKD kernel:     ok 20 - kmalloc_uaf2
Oct 29 18:55:02 dbg-LKD kernel:     ok 21 - kfree_via_page
Oct 29 18:55:02 dbg-LKD kernel:     ok 22 - kfree_via_phys
Oct 29 18:55:03 dbg-LKD kernel:     ok 23 - kmem_cache_oob
Oct 29 18:55:03 dbg-LKD kernel:     ok 24 - memcg_accounted_kmem_cache
Oct 29 18:55:03 dbg-LKD kernel:     ok 25 - kasan_global_oob
Oct 29 18:55:03 dbg-LKD kernel:     ok 26 - kasan_stack_oob
Oct 29 18:55:03 dbg-LKD kernel:     ok 27 - kasan_alloca_oob_left
Oct 29 18:55:03 dbg-LKD kernel:     ok 28 - kasan_alloca_oob_right
Oct 29 18:55:03 dbg-LKD kernel:     ok 29 - ksize_unpoisons_memory
Oct 29 18:55:03 dbg-LKD kernel:     ok 30 - kmem_cache_double_free
Oct 29 18:55:03 dbg-LKD kernel:     ok 31 - kmem_cache_invalid_free
Oct 29 18:55:03 dbg-LKD kernel:     ok 32 - kasan_memchr
Oct 29 18:55:03 dbg-LKD kernel:     ok 33 - kasan_memcmp
Oct 29 18:55:03 dbg-LKD kernel:     ok 34 - kasan_strings
Oct 29 18:55:04 dbg-LKD kernel:     ok 35 - kasan_bitops_generic
Oct 29 18:55:04 dbg-LKD kernel:     ok 36 - kasan_bitops_tags
Oct 29 18:55:04 dbg-LKD kernel:     ok 37 - kmalloc_double_kzfree
Oct 29 18:55:04 dbg-LKD kernel:     ok 38 - vmalloc_oob
```


Stack memory start
frame 1: foo1()
frame 2: bar1()
frame 3: foo2()
frame 4: bar2() <stale frame>
high KVA (kx+0x2000)
1
2
function call: decrease SP
Kernel stack
3
4
Page 0 of kernel stack
Stack Pointer register (SP)
5
function return: increase SP
downward-growing stack
Kernel stack size: 4 pages (64-bit) 2 pages (32-bit)
Page 1 of kernel stack
low KVA (kx)
KVA: kernel virtual address

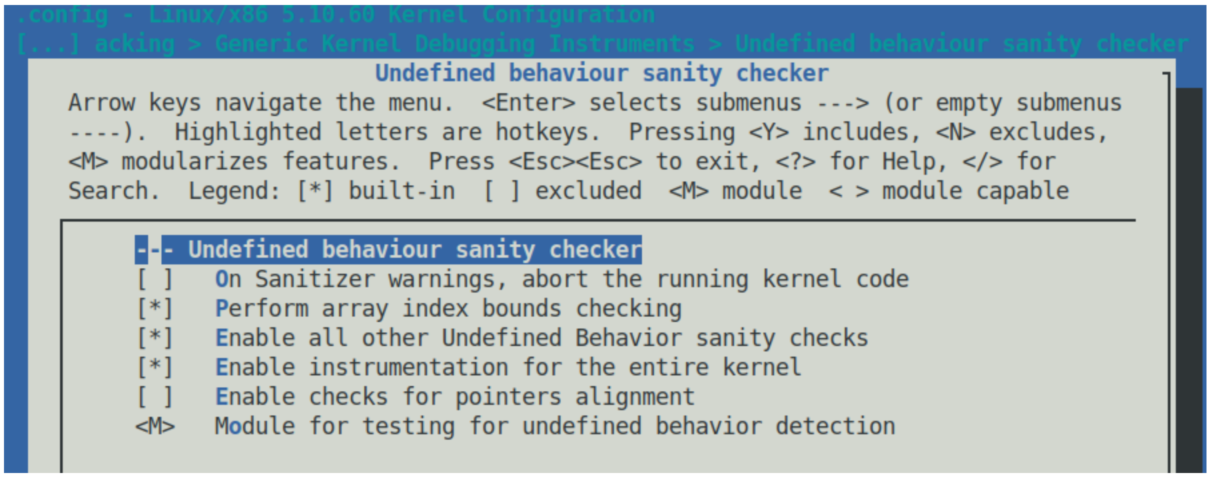


```
$ lsmod |grep test_kmembugs
test_kmembugs          61440  0
$ sudo ./run_tests
Debugfs file: /sys/kernel/debug/test_kmembugs/lkd_dbgfs_run_testcase

Generic KASAN: enabled
UBSAN: enabled
KMEMLEAK: enabled

Select testcase to run:
1  Uninitialized Memory Read - UMR
2  Use After Return - UAR

Memory leakage
3.1  simple memory leakage testcase1
3.2  simple memory leakage testcase2 - caller to free memory
3.3  simple memory leakage testcase3 - memleak in interrupt ctx

OOB accesses on static (compile-time) global memory + on stack local memory
4.1  Read  (right) overflow
4.2  Write (right) overflow
4.3  Read  (left) underflow
4.4  Write (left) underflow

OOB accesses on dynamic (kmalloc-ed) memory
5.1  Read  (right) overflow
5.2  Write (right) overflow
5.3  Read  (left) underflow
5.4  Write (left) underflow

6  Use After Free - UAF
7  Double-free

UBSAN arithmetic UB testcases
8.1  add overflow
8.2  sub overflow
8.3  mul overflow
8.4  negate overflow
8.5  shift OOB
8.6  OOB
8.7  load invalid value
8.8  misaligned access
8.9  object size mismatch

9  copy_[to|from]_user*() tests
10 UMR on slab (SLUB) memory

(Type in the testcase number to run):
2
Running testcase "2" via test module now...
[89638.348632] testcase to run: 2
[89638.350942] test_kmembugs:uar(): testcase 2: UAR:
[89638.352918] testcase 2: UAR: res1 = "<whoops, it's NULL; UAR!>"
$
```

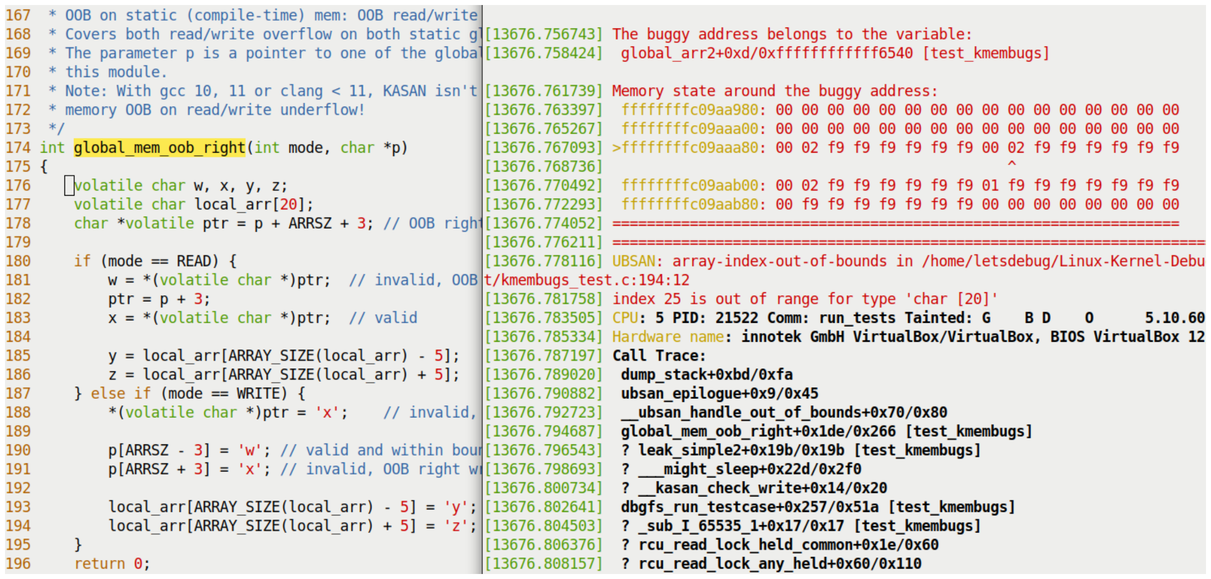


```
 */
int global_mem_oob_left(int mode, char *p)
{
    volatile char w, x, y, z;
    volatile char local_arr[20];
    char *volatile ptr = p - 3; // left OOB

    if (mode == READ) {
        /* Interesting: this OOB access isn't
        w = *(volatile char *)ptr;  // invalid

        /* ... but these OOB accesses are caug
         * We conclude that *only* the index-b
         * And, KASAN compiled with clang 11 c
         */
        x = p[-3];  // invalid, OOB left read

        y = local_arr[-5];  // invalid, not wi
        z = local_arr[5];   // valid, within b
    } else if (mode == WRITE) {
        /* Interesting: this OOB access isn't
        *(volatile char *)ptr = 'w';
```

```
9  copy_[to|from]_user*() tests
10 UMR on slab (SLUB) memory

(Type in the testcase number to run):
4.4
Running testcase "4.4" via test module now...
[13372.544725] testcase to run: 4.4
[13372.553282] ==================================================================
[13372.562448] BUG: KASAN: global-out-of-bounds in global_mem_oob_left+0x172/0x267 [test_kmembugs]
[13372.571100] Write of size 1 at addr ffffffffc09aaabd by task run_tests/21489

[13372.581341] CPU: 0 PID: 21489 Comm: run_tests Tainted: G    B D    O      5.10.60-dbg02-gcc #17
[13372.585154] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[13372.588567] Call Trace:
[13372.591665]  dump_stack+0xbd/0xfa
[13372.594362]  print_address_description.constprop.0.cold+0x5/0x4db
[13372.597040]  ? trace_preempt_off+0x2a/0xf0
[13372.599451]  ? global_mem_oob_left+0x172/0x267 [test_kmembugs]
[13372.601856]  kasan_report.cold+0x37/0x7c
[13372.603928]  ? global_mem_oob_left+0x172/0x267 [test_kmembugs]
[13372.605993]  __asan_store1+0x6d/0x70
[13372.607932]  global_mem_oob_left+0x172/0x267 [test_kmembugs]
```

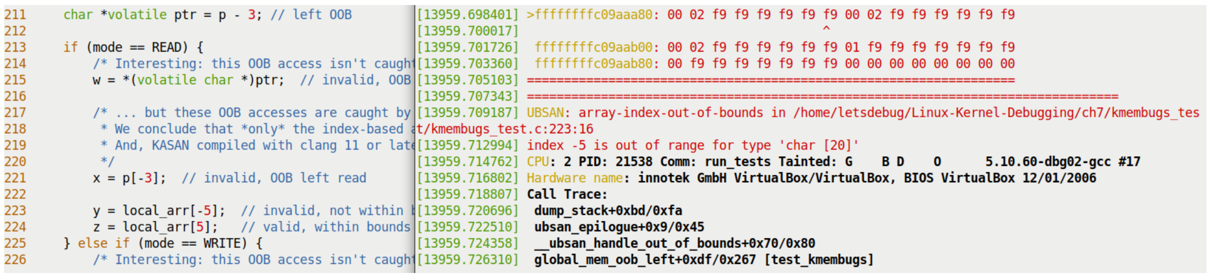


```
.config - Linux/x86 5.10.60 Kernel Configuration
[...] acking > Generic Kernel Debugging Instruments > Undefined behaviour sanity checker
                     Undefined behaviour sanity checker
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus
  ----).  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes,
  <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for
  Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

        --- Undefined behaviour sanity checker
        [ ]   On Sanitizer warnings, abort the running kernel code
        [*]   Perform array index bounds checking
        [*]   Enable all other Undefined Behavior sanity checks
        [*]   Enable instrumentation for the entire kernel
        [ ]   Enable checks for pointers alignment
        <M>   Module for testing for undefined behavior detection
```



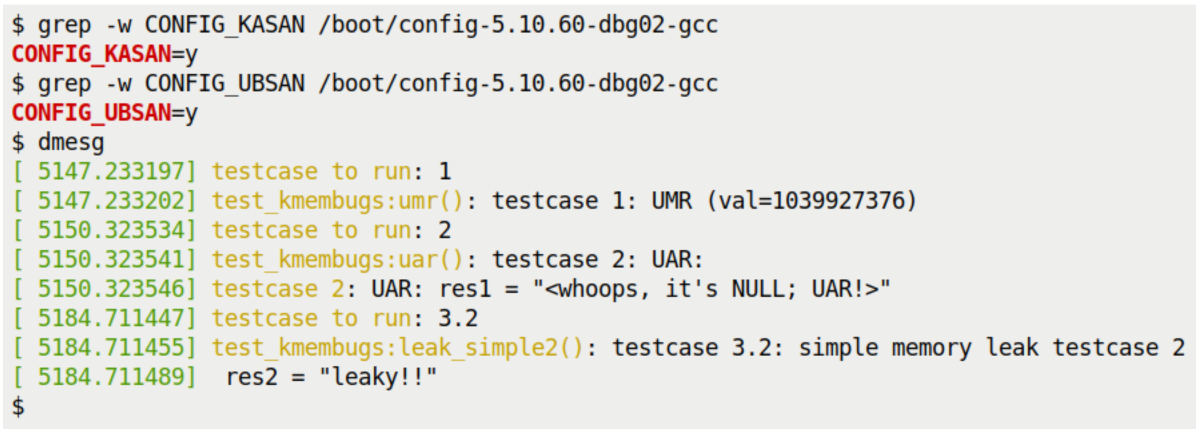
```
167  * OOB on static (compile-time) mem: OOB read/write
168  * Covers both read/write overflow on both static g
169  * The parameter p is a pointer to one of the globa
170  * this module.
171  * Note: With gcc 10, 11 or clang < 11, KASAN isn't
172  * memory OOB on read/write underflow!
173  */
174 int global_mem_oob_right(int mode, char *p)
175 {
176     volatile char w, x, y, z;
177     volatile char local_arr[20];
178     char *volatile ptr = p + ARRSZ + 3; // OOB righ
179 
180     if (mode == READ) {
181         w = *(volatile char *)ptr;  // invalid, OOB
182         ptr = p + 3;
183         x = *(volatile char *)ptr;  // valid
184 
185         y = local_arr[ARRAY_SIZE(local_arr) - 5];
186         z = local_arr[ARRAY_SIZE(local_arr) + 5];
187     } else if (mode == WRITE) {
188         *(volatile char *)ptr = 'x';    // invalid,
189 
190         p[ARRSZ - 3] = 'w'; // valid and within bou
191         p[ARRSZ + 3] = 'x'; // invalid, OOB right w
192 
193         local_arr[ARRAY_SIZE(local_arr) - 5] = 'y';
194         local_arr[ARRAY_SIZE(local_arr) + 5] = 'z';
195     }
196     return 0;
```

```
[13676.756743] The buggy address belongs to the variable:
[13676.758424]  global_arr2+0xd/0xfffffffffff6540 [test_kmembugs]

[13676.761739] Memory state around the buggy address:
[13676.763397]  ffffffffc09aa980: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[13676.765267]  ffffffffc09aaa00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[13676.767093] >ffffffffc09aaa80: 00 02 f9 f9 f9 f9 f9 f9 00 02 f9 f9 f9 f9 f9 f9
[13676.768736]                                               ^
[13676.770492]  ffffffffc09aab00: 00 02 f9 f9 f9 f9 f9 f9 01 f9 f9 f9 f9 f9 f9 f9
[13676.772293]  ffffffffc09aab80: 00 f9 f9 f9 f9 f9 f9 f9 00 00 00 00 00 00 00 00
[13676.774052] ==================================================================
[13676.776211] ====================================================================
[13676.778116] UBSAN: array-index-out-of-bounds in /home/letsdebug/Linux-Kernel-Debu
t/kmembugs_test.c:194:12
[13676.781758] index 25 is out of range for type 'char [20]'
[13676.783505] CPU: 5 PID: 21522 Comm: run_tests Tainted: G    B D    O      5.10.60
[13676.785334] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12
[13676.787197] Call Trace:
[13676.789020]  dump_stack+0xbd/0xfa
[13676.790882]  ubsan_epilogue+0x9/0x45
[13676.792723]  __ubsan_handle_out_of_bounds+0x70/0x80
[13676.794687]  global_mem_oob_right+0x1de/0x266 [test_kmembugs]
[13676.796543]  ? leak_simple2+0x19b/0x19b [test_kmembugs]
[13676.798693]  ? ___might_sleep+0x22d/0x2f0
[13676.800734]  ? __kasan_check_write+0x14/0x20
[13676.802641]  dbgfs_run_testcase+0x257/0x51a [test_kmembugs]
[13676.804503]  ? _sub_I_65535_1+0x17/0x17 [test_kmembugs]
[13676.806376]  ? rcu_read_lock_held_common+0x1e/0x60
[13676.808157]  ? rcu_read_lock_any_held+0x60/0x110
```

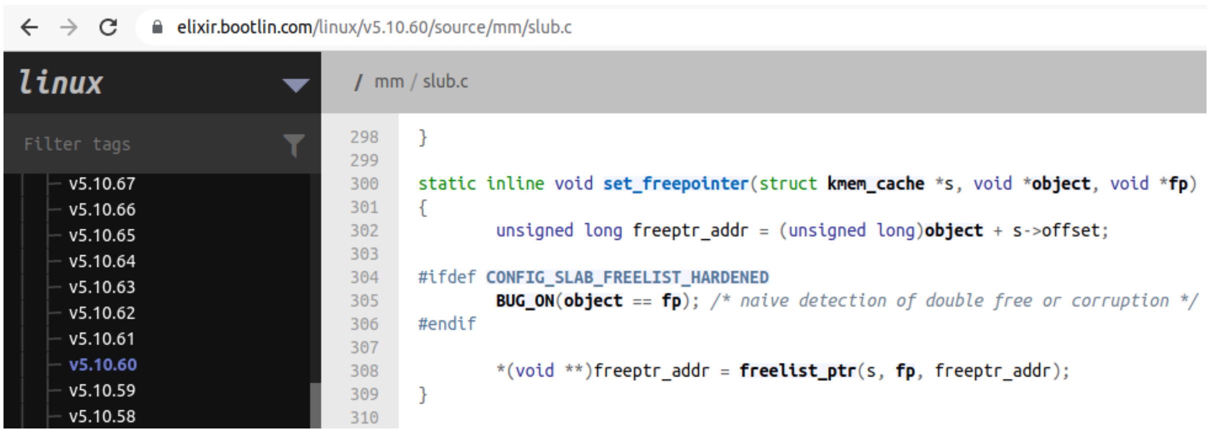


```
211     char *volatile ptr = p - 3; // left OOB
212
213     if (mode == READ) {
214         /* Interesting: this OOB access isn't caught
215         w = *(volatile char *)ptr;  // invalid, OOB
216
217         /* ... but these OOB accesses are caught by
218          * We conclude that *only* the index-based a
219          * And, KASAN compiled with clang 11 or late
220          */
221         x = p[-3];  // invalid, OOB left read
222
223         y = local_arr[-5];  // invalid, not within b
224         z = local_arr[5];   // valid, within bounds
225     } else if (mode == WRITE) {
226         /* Interesting: this OOB access isn't caught
```

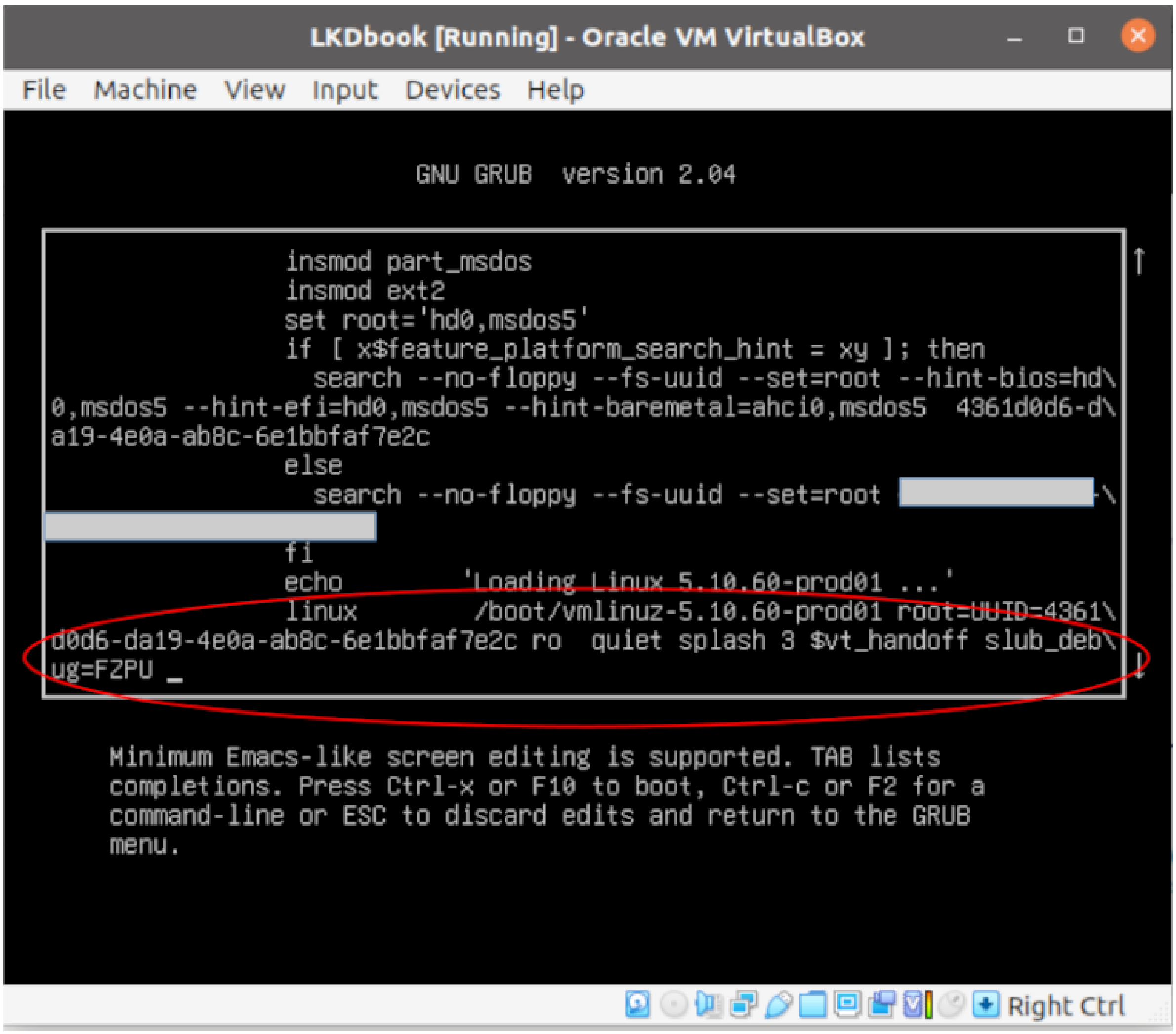
```
[13959.698401] >ffffffffc09aaa80: 00 02 f9 f9 f9 f9 f9 f9 00 02 f9 f9 f9 f9 f9 f9
[13959.700017]                                      ^
[13959.701726]  ffffffffc09aab00: 00 02 f9 f9 f9 f9 f9 f9 01 f9 f9 f9 f9 f9 f9 f9
[13959.703360]  ffffffffc09aab80: 00 f9 f9 f9 f9 f9 f9 f9 00 00 00 00 00 00 00 00
[13959.705103] ==================================================================
[13959.707343] ================================================================================
[13959.709187] UBSAN: array-index-out-of-bounds in /home/letsdebug/Linux-Kernel-Debugging/ch7/kmembugs_tes
t/kmembugs_test.c:223:16
[13959.712994] index -5 is out of range for type 'char [20]'
[13959.714762] CPU: 2 PID: 21538 Comm: run_tests Tainted: G    B D    O      5.10.60-dbg02-gcc #17
[13959.716802] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[13959.718807] Call Trace:
[13959.720696]  dump_stack+0xbd/0xfa
[13959.722510]  ubsan_epilogue+0x9/0x45
[13959.724358]  __ubsan_handle_out_of_bounds+0x70/0x80
[13959.726310]  global_mem_oob_left+0xdf/0x267 [test_kmembugs]
```



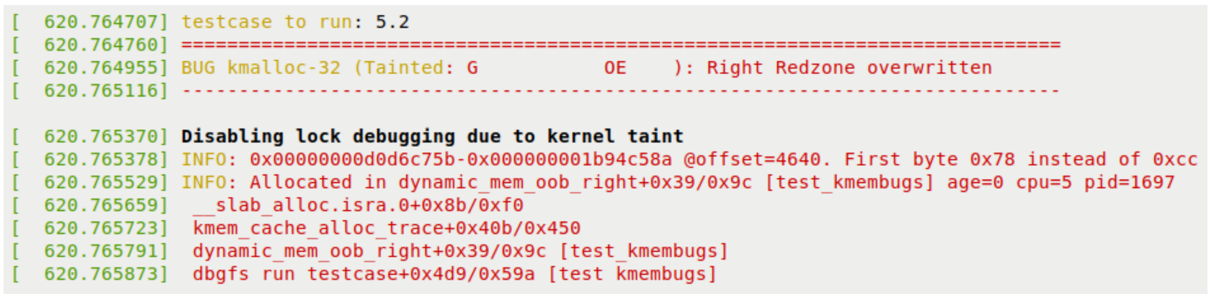
```
$ grep -w CONFIG_KASAN /boot/config-5.10.60-dbg02-gcc
CONFIG_KASAN=y
$ grep -w CONFIG_UBSAN /boot/config-5.10.60-dbg02-gcc
CONFIG_UBSAN=y
$ dmesg
[ 5147.233197] testcase to run: 1
[ 5147.233202] test_kmembugs:umr(): testcase 1: UMR (val=1039927376)
[ 5150.323534] testcase to run: 2
[ 5150.323541] test_kmembugs:uar(): testcase 2: UAR:
[ 5150.323546] testcase 2: UAR: res1 = "<whoops, it's NULL; UAR!>"
[ 5184.711447] testcase to run: 3.2
[ 5184.711455] test_kmembugs:leak_simple2(): testcase 3.2: simple memory leak testcase 2
[ 5184.711489]  res2 = "leaky!!"
$
```



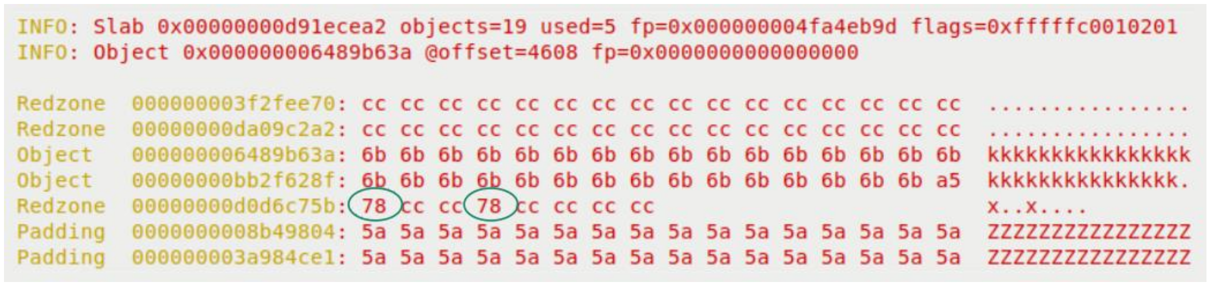
```
}

static inline void set_freepointer(struct kmem_cache *s, void *object, void *fp)
{
        unsigned long freeptr_addr = (unsigned long)object + s->offset;

#ifdef CONFIG_SLAB_FREELIST_HARDENED
        BUG_ON(object == fp); /* naive detection of double free or corruption */
#endif

        *(void **)freeptr_addr = freelist_ptr(s, fp, freeptr_addr);
}
```



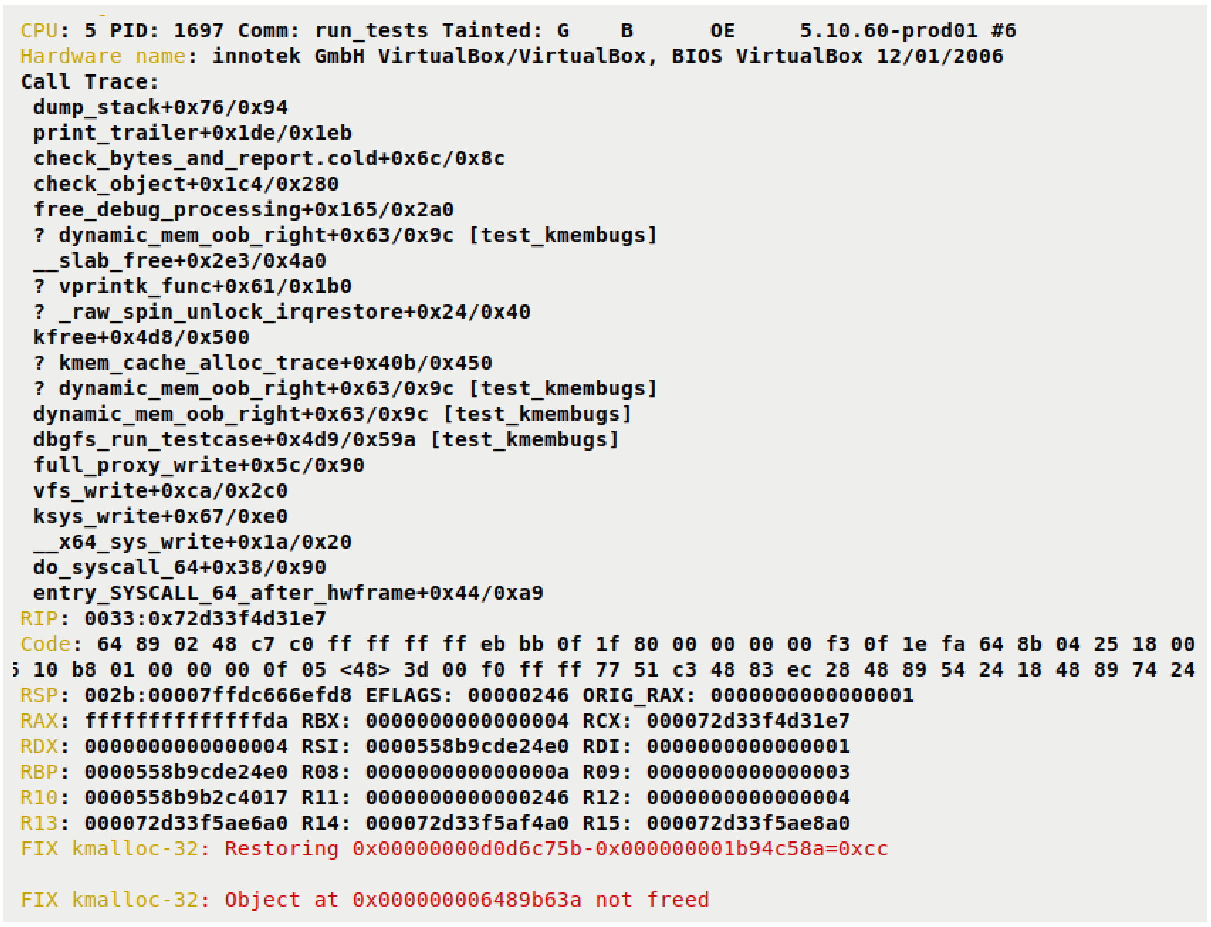
```
                         GNU GRUB  version 2.04

                insmod part_msdos
                insmod ext2
                set root='hd0,msdos5'
                if [ x$feature_platform_search_hint = xy ]; then
                  search --no-floppy --fs-uuid --set=root --hint-bios=hd\
0,msdos5 --hint-efi=hd0,msdos5 --hint-baremetal=ahci0,msdos5  4361d0d6-d\
a19-4e0a-ab8c-6e1bbfaf7e2c
                else
                  search --no-floppy --fs-uuid --set=root [redacted]\
[redacted]
                fi
                echo         'Loading Linux 5.10.60-prod01 ...'
                linux        /boot/vmlinuz-5.10.60-prod01 root=UUID=4361\
d0d6-da19-4e0a-ab8c-6e1bbfaf7e2c ro  quiet splash 3 $vt_handoff slub_deb\
ug=FZPU _

    Minimum Emacs-like screen editing is supported. TAB lists
    completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a
    command-line or ESC to discard edits and return to the GRUB
    menu.
```



```
[  620.764707] testcase to run: 5.2
[  620.764760] =============================================================================
[  620.764955] BUG kmalloc-32 (Tainted: G           OE    ): Right Redzone overwritten
[  620.765116] -----------------------------------------------------------------------------

[  620.765370] Disabling lock debugging due to kernel taint
[  620.765378] INFO: 0x00000000d0d6c75b-0x000000001b94c58a @offset=4640. First byte 0x78 instead of 0xcc
[  620.765529] INFO: Allocated in dynamic_mem_oob_right+0x39/0x9c [test_kmembugs] age=0 cpu=5 pid=1697
[  620.765659]  __slab_alloc.isra.0+0x8b/0xf0
[  620.765723]  kmem_cache_alloc_trace+0x40b/0x450
[  620.765791]  dynamic_mem_oob_right+0x39/0x9c [test_kmembugs]
[  620.765873]  dbgfs run testcase+0x4d9/0x59a [test kmembugs]
```



```
INFO: Slab 0x00000000d91ecea2 objects=19 used=5 fp=0x000000004fa4eb9d flags=0xfffffc0010201
INFO: Object 0x000000006489b63a @offset=4608 fp=0x0000000000000000

Redzone  000000003f2fee70: cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc  ................
Redzone  00000000da09c2a2: cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc  ................
Object   000000006489b63a: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Object   00000000bb2f628f: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b a5  kkkkkkkkkkkkkkk.
Redzone  00000000d0d6c75b: 78 cc cc 78 cc cc cc cc                          x..x....
Padding  000000008b49804: 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a  ZZZZZZZZZZZZZZZZ
Padding  000000003a984ce1: 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a  ZZZZZZZZZZZZZZZZ
```

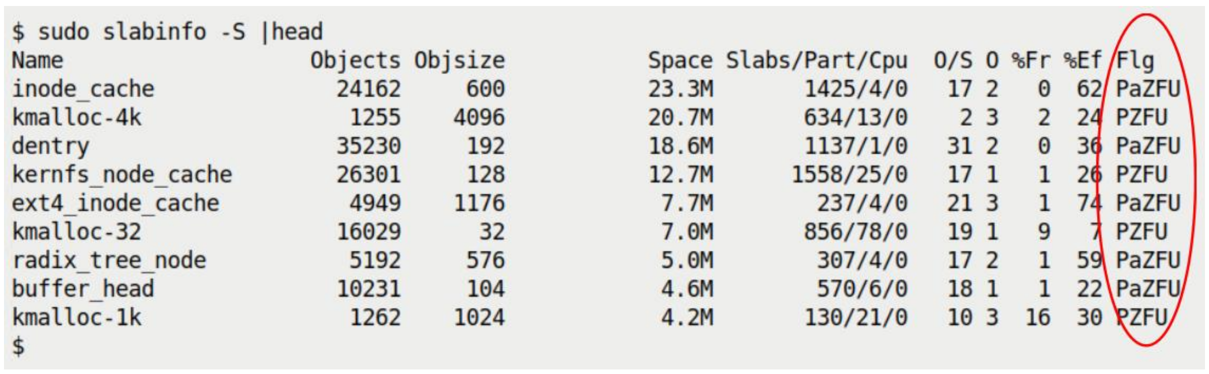


```
CPU: 5 PID: 1697 Comm: run_tests Tainted: G    B      OE     5.10.60-prod01 #6
Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
Call Trace:
 dump_stack+0x76/0x94
 print_trailer+0x1de/0x1eb
 check_bytes_and_report.cold+0x6c/0x8c
 check_object+0x1c4/0x280
 free_debug_processing+0x165/0x2a0
 ? dynamic_mem_oob_right+0x63/0x9c [test_kmembugs]
 __slab_free+0x2e3/0x4a0
 ? vprintk_func+0x61/0x1b0
 ? _raw_spin_unlock_irqrestore+0x24/0x40
 kfree+0x4d8/0x500
 ? kmem_cache_alloc_trace+0x40b/0x450
 ? dynamic_mem_oob_right+0x63/0x9c [test_kmembugs]
 dynamic_mem_oob_right+0x63/0x9c [test_kmembugs]
 dbgfs_run_testcase+0x4d9/0x59a [test_kmembugs]
 full_proxy_write+0x5c/0x90
 vfs_write+0xca/0x2c0
 ksys_write+0x67/0xe0
 __x64_sys_write+0x1a/0x20
 do_syscall_64+0x38/0x90
 entry_SYSCALL_64_after_hwframe+0x44/0xa9
RIP: 0033:0x72d33f4d31e7
Code: 64 89 02 48 c7 c0 ff ff ff ff eb bb 0f 1f 80 00 00 00 00 f3 0f 1e fa 64 8b 04 25 18 00
5 10 b8 01 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 51 c3 48 83 ec 28 48 89 54 24 18 48 89 74 24
RSP: 002b:00007ffdc666efd8 EFLAGS: 00000246 ORIG_RAX: 0000000000000001
RAX: ffffffffffffffda RBX: 0000000000000004 RCX: 000072d33f4d31e7
RDX: 0000000000000004 RSI: 0000558b9cde24e0 RDI: 0000000000000001
RBP: 0000558b9cde24e0 R08: 000000000000000a R09: 0000000000000003
R10: 0000558b9b2c4017 R11: 0000000000000246 R12: 0000000000000004
R13: 000072d33f5ae6a0 R14: 000072d33f5af4a0 R15: 000072d33f5ae8a0
FIX kmalloc-32: Restoring 0x00000000d0d6c75b-0x000000001b94c58a=0xcc

FIX kmalloc-32: Object at 0x000000006489b63a not freed
```

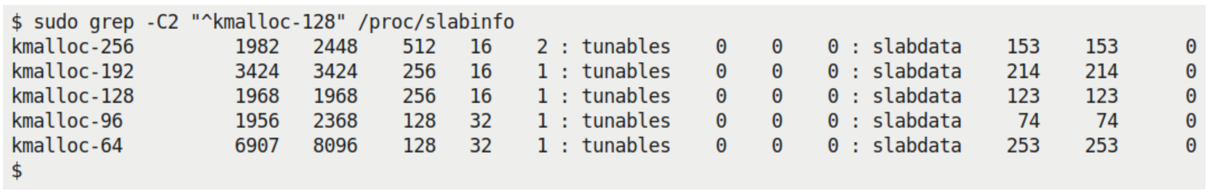
```
$ sudo slabinfo --help
slabinfo 4/15/2011. (c) 2007 sgi/(c) 2011 Linux Foundation.

slabinfo [-aABDefhilLnoPrsStTUvXz1] [N=K] [-dafzput] [slab-regexp]
-a|--aliases           Show aliases
-A|--activity          Most active slabs first
-B|--Bytes             Show size in bytes
-D|--display-active    Switch line format to activity
-e|--empty             Show empty slabs
-f|--first-alias       Show first alias
-h|--help              Show usage information
-i|--inverted          Inverted list
-l|--slabs             Show slabs
-L|--Loss              Sort by loss
-n|--numa              Show NUMA information
-N|--lines=K           Show the first K slabs
-o|--ops               Show kmem_cache_ops
-P|--partial            Sort by number of partial slabs
-r|--report            Detailed report on single slabs
-s|--shrink            Shrink slabs
-S|--Size              Sort by size
-t|--tracking          Show alloc/free information
-T|--Totals            Show summary information
-U|--Unreclaim         Show unreclaimable slabs only
-v|--validate          Validate slabs
-X|--Xtotals           Show extended summary information
-z|--zero              Include empty slabs
-1|--1ref              Single reference

-d  | --debug          Switch off all debug options
-da | --debug=a        Switch on all debug options (--debug=FZPU)

-d[afzput] | --debug=[afzput]
    f | F              Sanity Checks (SLAB_CONSISTENCY_CHECKS)
    z | Z              Redzoning
    p | P              Poisoning
    u | U              Tracking
    t | T              Tracing

Sorting options (--Loss, --Size, --Partial) are mutually exclusive
$
```

```
$ sudo slabinfo -S | head
Name                   Objects Objsize           Space Slabs/Part/Cpu  O/S O %Fr %Ef Flg
inode_cache              24726     600           15.5M      912/0/39    26 2   0  95 a
buffer_head             132015     104           13.8M     3369/0/16    39 0   0  99 a
ext4_inode_cache          7074    1176            8.5M      252/0/10    27 3   0  96 a
dentry                   40572     192            7.9M     1883/0/49    21 0   0  98 a
kmalloc-4k                1591    4096            6.5M      189/8/12     8 3   3  98
radix_tree_node           8603     576            5.0M      298/7/13    28 2   2  97 a
kernfs_node_cache        30144     128            3.8M      899/0/43    32 0   0 100
kmalloc-512               5040     512            2.5M      282/0/33    16 1   0 100
filp                      8816     256            2.2M      501/0/51    16 0   0  99 A
$ _
```

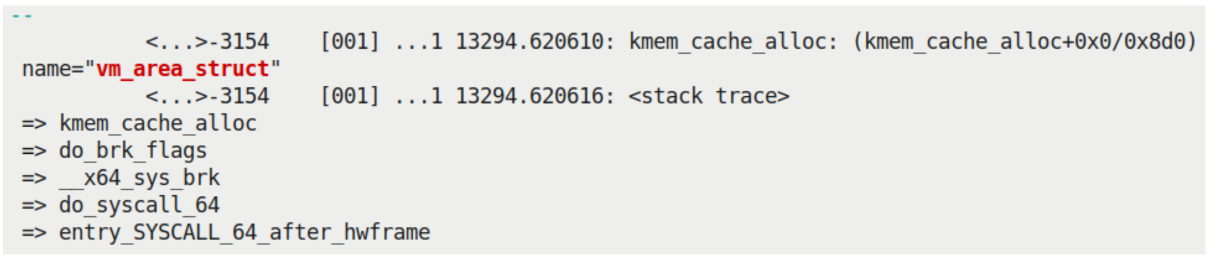
```
$ sudo slabinfo -X
[sudo] password for letsdebug:
Slabcache Totals
----------------
Slabcaches :              216   Aliases  :           0->0    Active:     133
Memory used:         90710016   # Loss   :        2548968    MRatio:      2%
# Objects  :           401015   # PartObj:           1444    ORatio:      0%

Per Cache         Average              Min              Max            Total
---------------------------------------------------------------------------
#Objects             3015               10           132132           401015
#Slabs                 88                1             3388            11833
#PartSlab               0                0               31              101
%PartSlab              0%               0%              38%               0%
PartObjs                0                0              670             1444
% PartObj              0%               0%              23%               0%
Memory             682030             4096         15581184         90710016
Used               662865             3072         14835600         88161048
Loss                19165                0           745584          2548968

Per Object        Average              Min              Max
---------------------------------------------------------
Memory                221                8             8192
User                  219                8             8192
Loss                    1                0               64

Slabs sorted by size
--------------------
Name                   Objects Objsize           Space Slabs/Part/Cpu  O/S O %Fr %Ef Flg
inode_cache              24726     600        15581184      912/0/39    26 2   0  95 a

Slabs sorted by loss
--------------------
Name                   Objects Objsize            Loss Slabs/Part/Cpu  O/S O %Fr %Ef Flg
inode_cache              24726     600          745584      912/0/39    26 2   0  95 a

Slabs sorted by number of partial slabs
---------------------------------------
Name                   Objects Objsize           Space Slabs/Part/Cpu  O/S O %Fr %Ef Flg
anon_vma                  2970      80          331776      40/31/41    46 0  38  71

$
```



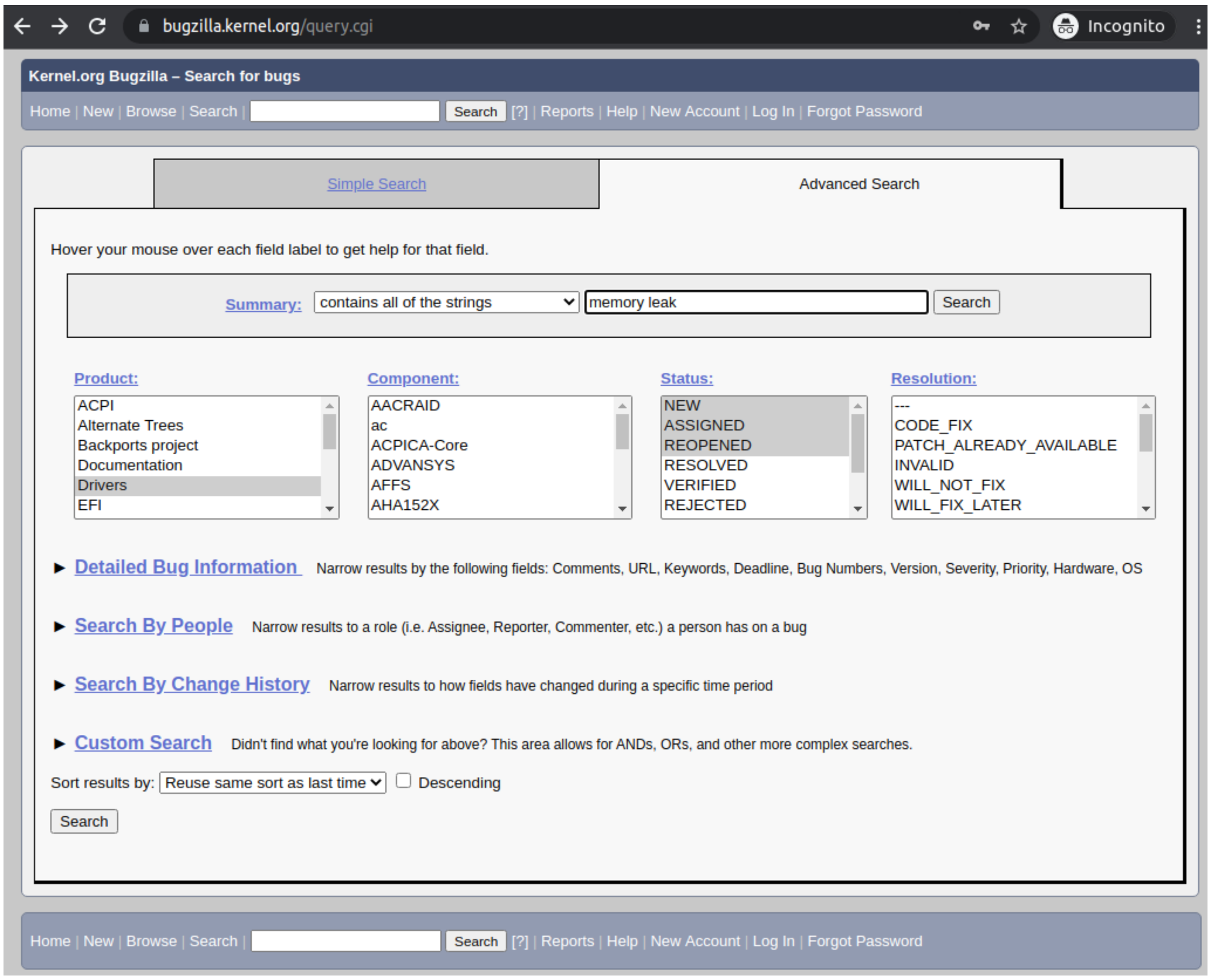
```
$ sudo slabinfo -S |head
Name                   Objects Objsize           Space Slabs/Part/Cpu  O/S O %Fr %Ef Flg
inode_cache              24162     600           23.3M      1425/4/0    17 2   0  62 PaZFU
kmalloc-4k                1255    4096           20.7M     634/13/0     2 3   2  24 PZFU
dentry                   35230     192           18.6M      1137/1/0    31 2   0  36 PaZFU
kernfs_node_cache        26301     128           12.7M    1558/25/0    17 1   1  26 PZFU
ext4_inode_cache          4949    1176            7.7M       237/4/0    21 3   1  74 PaZFU
kmalloc-32               16029      32            7.0M     856/78/0    19 1   9   7 PZFU
radix_tree_node           5192     576            5.0M       307/4/0    17 2   1  59 PaZFU
buffer_head              10231     104            4.6M       570/6/0    18 1   1  22 PaZFU
kmalloc-1k                1262    1024            4.2M     130/21/0    10 3  16  30 PZFU
$
```



```
$ sudo grep -C2 "^kmalloc-128" /proc/slabinfo
kmalloc-256         1982   2448    512   16    2 : tunables    0    0    0 : slabdata    153    153      0
kmalloc-192         3424   3424    256   16    1 : tunables    0    0    0 : slabdata    214    214      0
kmalloc-128         1968   1968    256   16    1 : tunables    0    0    0 : slabdata    123    123      0
kmalloc-96          1956   2368    128   32    1 : tunables    0    0    0 : slabdata     74     74      0
kmalloc-64          6907   8096    128   32    1 : tunables    0    0    0 : slabdata    253    253      0
$
```



```
--
          <...>-3154    [001] ...1 13294.620610: kmem_cache_alloc: (kmem_cache_alloc+0x0/0x8d0)
 name="vm_area_struct"
          <...>-3154    [001] ...1 13294.620616: <stack trace>
 => kmem_cache_alloc
 => do_brk_flags
 => __x64_sys_brk
 => do_syscall_64
 => entry_SYSCALL_64_after_hwframe
```



bugzilla.kernel.org/query.cgi

Kernel.org Bugzilla – Search for bugs

Home | New | Browse | Search | [Search] [?] | Reports | Help | New Account | Log In | Forgot Password

Simple Search | Advanced Search

Hover your mouse over each field label to get help for that field.

Summary: contains all of the strings | memory leak | Search

Product:	Component:	Status:	Resolution:
ACPI	AACRAID	NEW	---
Alternate Trees	ac	ASSIGNED	CODE_FIX
Backports project	ACPICA-Core	REOPENED	PATCH_ALREADY_AVAILABLE
Documentation	ADVANSYS	RESOLVED	INVALID
Drivers	AFFS	VERIFIED	WILL_NOT_FIX
EFI	AHA152X	REJECTED	WILL_FIX_LATER

▶ Detailed Bug Information Narrow results by the following fields: Comments, URL, Keywords, Deadline, Bug Numbers, Version, Severity, Priority, Hardware, OS

▶ Search By People Narrow results to a role (i.e. Assignee, Reporter, Commenter, etc.) a person has on a bug

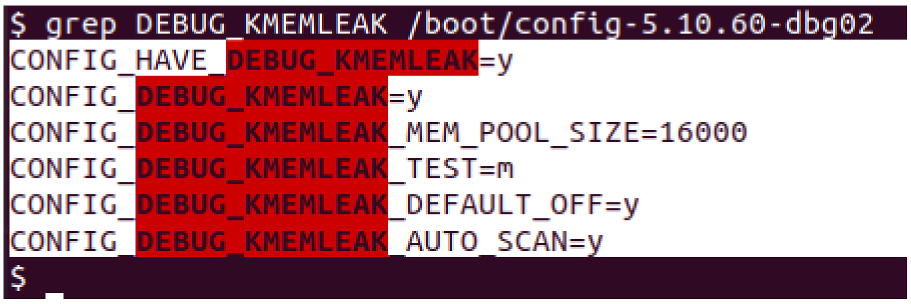
▶ Search By Change History Narrow results to how fields have changed during a specific time period

▶ Custom Search Didn't find what you're looking for above? This area allows for ANDs, ORs, and other more complex searches.

Sort results by: Reuse same sort as last time ☐ Descending

Search

Home | New | Browse | Search | [Search] [?] | Reports | Help | New Account | Log In | Forgot Password



```
$ grep DEBUG_KMEMLEAK /boot/config-5.10.60-dbg02
CONFIG_HAVE_DEBUG_KMEMLEAK=y
CONFIG_DEBUG_KMEMLEAK=y
CONFIG_DEBUG_KMEMLEAK_MEM_POOL_SIZE=16000
CONFIG_DEBUG_KMEMLEAK_TEST=m
CONFIG_DEBUG_KMEMLEAK_DEFAULT_OFF=y
CONFIG_DEBUG_KMEMLEAK_AUTO_SCAN=y
$ 
```

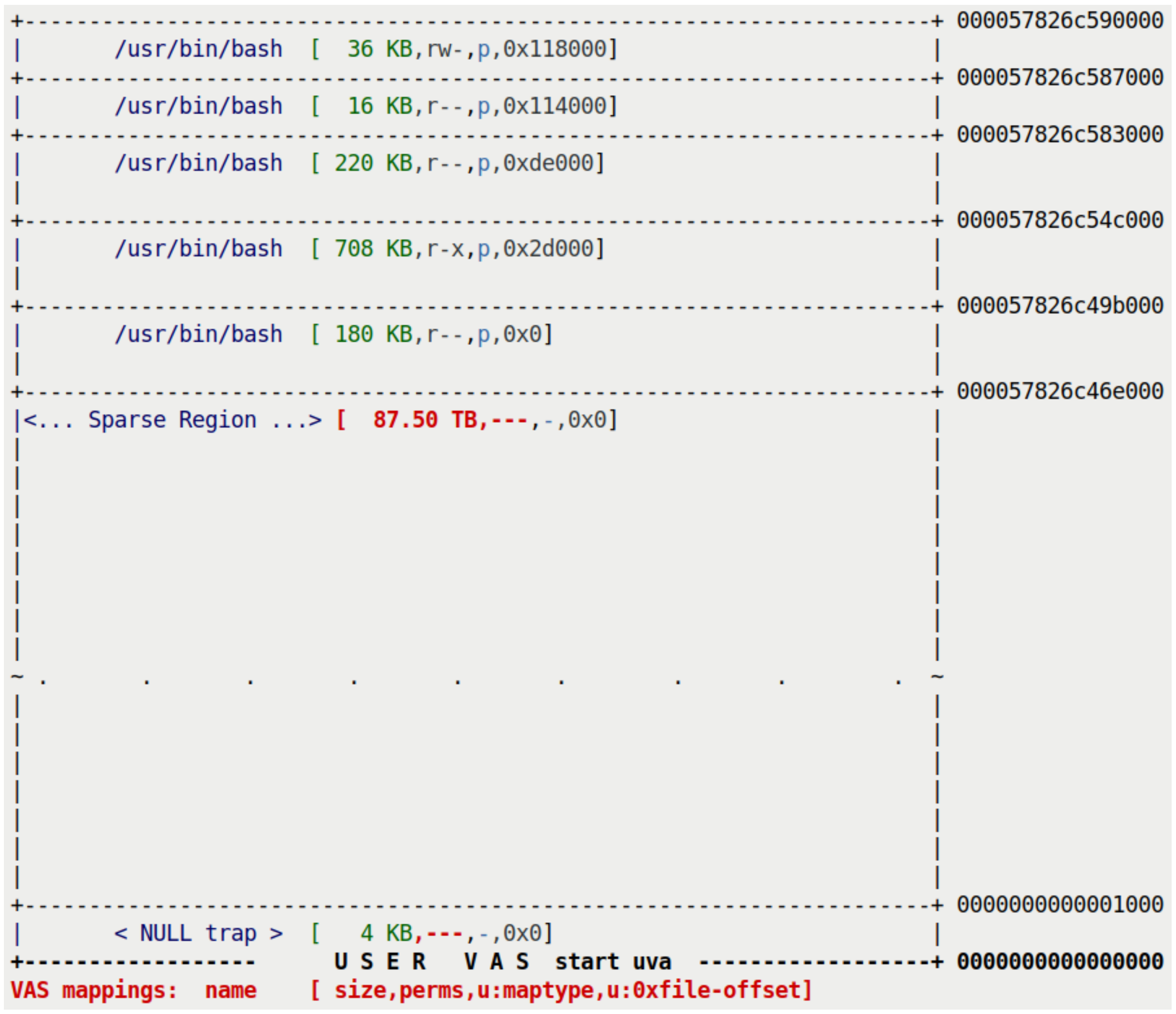
```
$ journalctl --output=short-unix -k |grep -iC2 "kmemleak"
1637844902.306232 dbg-LKD kernel: random: get_random_u64 called from __kmem_cache_create+0x2f/0x500 wit
h crng_init=0
1637844902.306303 dbg-LKD kernel: SLUB: HWalign=64, Order=0-3, MinObjects=0, CPUs=6, Nodes=1
1637844902.306367 dbg-LKD kernel: kmemleak: Kernel memory leak detector disabled
1637844902.306441 dbg-LKD kernel: Kernel/User page tables isolation: enabled
1637844902.306506 dbg-LKD kernel: ftrace: allocating 44433 entries in 174 pages
--
1637844902.629853 dbg-LKD kernel: calling  split_huge_pages_debugfs+0x0/0x29 @ 1
1637844902.629942 dbg-LKD kernel: initcall split_huge_pages_debugfs+0x0/0x29 returned 0 after 23 usecs
1637844902.630024 dbg-LKD kernel: calling  kmemleak_late_init+0x0/0xa1 @ 1
1637844902.630093 dbg-LKD kernel: initcall kmemleak_late_init+0x0/0xa1 returned -12 after 30 usecs
1637844902.630159 dbg-LKD kernel: calling  check_early_ioremap_leak+0x0/0x9e @ 1
1637844902.630225 dbg-LKD kernel: initcall check_early_ioremap_leak+0x0/0x9e returned 0 after 872 usecs
```

```
$ sudo cat /sys/kernel/debug/kmemleak
unreferenced object 0xffff8880127f8000 (size 2048):
  comm "run_tests", pid 5498, jiffies 4296684850 (age 84.737s)
  hex dump (first 32 bytes):
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
  backtrace:
    [<00000000c0b84cb6>] slab_post_alloc_hook+0x78/0x5b0
    [<00000000f76c1d8d>] kmem_cache_alloc_trace+0x16b/0x370
    [<00000000896eb2a4>] leak_simple1+0x45/0x90 [test_kmembugs]
    [<000000000fca301f>] dbgfs_run_testcase+0x1c7/0x51a [test_kmembugs]
    [<00000000f0fd1df8>] full_proxy_write+0xaf/0xe0
    [<000000000d54f8ef>] vfs_write+0x148/0x500
    [<000000007f738be9>] ksys_write+0xd9/0x180
    [<000000001fce737f>] __x64_sys_write+0x43/0x50
    [<000000001a646102>] do_syscall_64+0x38/0x90
    [<0000000024b0a009>] entry_SYSCALL_64_after_hwframe+0x44/0xa9
unreferenced object 0xffffc90000065000 (size 8192):
  comm "run_tests", pid 5498, jiffies 4296684851 (age 84.734s)
  hex dump (first 32 bytes):
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
  backtrace:
    [<000000001fb65f64>] __vmalloc_node_range+0x476/0x4f0
    [<00000000c80cce1d>] __vmalloc_node+0xa7/0xd0
    [<000000001fd83f6a>] vmalloc+0x21/0x30
    [<000000005e2eaf52>] leak_simple1+0x71/0x90 [test_kmembugs]
    [<000000000fca301f>] dbgfs_run_testcase+0x1c7/0x51a [test_kmembugs]
    [<00000000f0fd1df8>] full_proxy_write+0xaf/0xe0
    [<000000000d54f8ef>] vfs_write+0x148/0x500
    [<000000007f738be9>] ksys_write+0xd9/0x180
    [<000000001fce737f>] __x64_sys_write+0x43/0x50
    [<000000001a646102>] do_syscall_64+0x38/0x90
    [<0000000024b0a009>] entry_SYSCALL_64_after_hwframe+0x44/0xa9
$
```

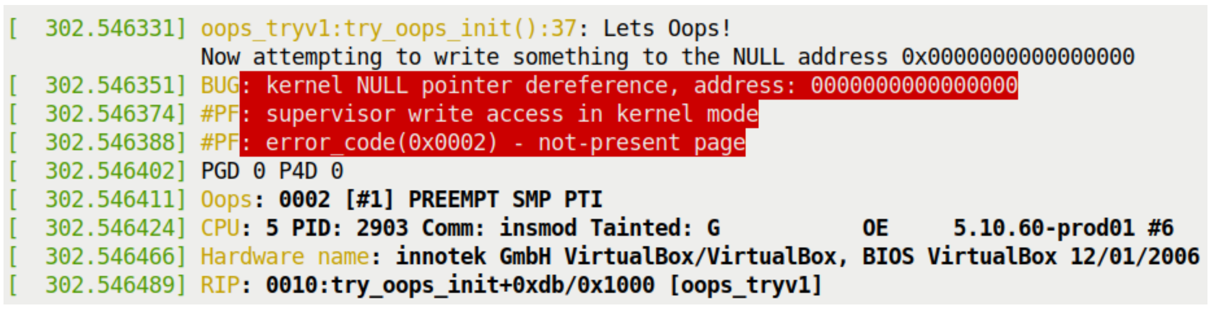
```
$ sudo sh -c "echo scan > /sys/kernel/debug/kmemleak" ; dmesg |tail
[34619.682989] test_kmembugs:kmembugs_test_init(): KASAN configured
[34619.684794] test_kmembugs:kmembugs_test_init(): CONFIG_UBSAN configured
[34619.686614] test_kmembugs:kmembugs_test_init(): CONFIG_DEBUG_KMEMLEAK configured
[34619.688443] debugfs file 1 <debugfs_mountpt>/test_kmembugs/lkd_dbgfs_run_testcase created
[34619.690270] debugfs entry initialized
[35412.528017] testcase to run: 3.3
[35412.530040] test_kmembugs:leak_simple3(): testcase 3.3: simple memory leak testcase 3
[35412.532750] test_kmembugs:irq_work_leaky(): 001)  run_tests :11781   |  d.h1   /* irq_work_leaky() */
[35412.537365] test_kmembugs:irq_work_leaky(): kzalloc(129) = 0xffff88803c1e0e00
[35438.671971] kmemleak: 1 new suspected memory leaks (see /sys/kernel/debug/kmemleak)
$
$ sudo cat /sys/kernel/debug/kmemleak
unreferenced object 0xffff88803c1e0e00 (size 192):
  comm "hardirq", pid 0, jiffies 4305500943 (age 34.834s)
  hex dump (first 32 bytes):
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
  backtrace:
    [<00000000c0b84cb6>] slab_post_alloc_hook+0x78/0x5b0
    [<00000000f76c1d8d>] kmem_cache_alloc_trace+0x16b/0x370
    [<000000002912ff8c>] irq_work_leaky+0x1f3/0x226 [test_kmembugs]
    [<00000000b094c375>] irq_work_single+0x8f/0xf0
    [<000000005a10cafa>] irq_work_run_list+0x52/0x70
    [<00000000e07f0913>] irq_work_run+0x6b/0x110
    [<000000006d70efc1>] __sysvec_irq_work+0x75/0x2b0
    [<0000000038851639>] asm_call_irq_on_stack+0x12/0x20
    [<000000006e1838aa>] sysvec_irq_work+0xc3/0xe0
    [<0000000043c320fa>] asm_sysvec_irq_work+0x12/0x20
    [<000000007864aefa>] native_write_msr+0x6/0x30
    [<0000000041cbb6ac>] x2apic_send_IPI_self+0x3c/0x50
    [<00000000b30d6970>] arch_irq_work_raise+0x5d/0x90
    [<00000000848d8ab3>] __irq_work_queue_local+0xf8/0x170
    [<00000000a3bb972c>] irq_work_queue+0x32/0x50
    [<000000005b977e7a>] leak_simple3+0x2f/0x31 [test_kmembugs]
$
```

```
[ 8825.985116] kmemleak: Kmemleak testing
[ 8825.985147] kmemleak: kmalloc(32) = 00000000cab708dd
[ 8825.985172] kmemleak: kmalloc(32) = 000000008d5c540a
[ 8825.985196] kmemleak: kmalloc(1024) = 000000006d719a53
[ 8825.985221] kmemleak: kmalloc(1024) = 00000000de599e5e
[ 8825.985247] kmemleak: kmalloc(2048) = 00000000b5e60406
[ 8825.985272] kmemleak: kmalloc(2048) = 000000000309c294
[ 8825.985299] kmemleak: kmalloc(4096) = 000000009200f455
[ 8825.985324] kmemleak: kmalloc(4096) = 000000001cfde96d
[ 8825.985555] kmemleak: vmalloc(64) = 00000000b7894b61
[ 8825.985672] kmemleak: vmalloc(64) = 00000000bbb401d6
[ 8825.985796] kmemleak: vmalloc(64) = 000000009c4e811f
[ 8825.985893] kmemleak: vmalloc(64) = 000000001e8fcc4a
[ 8825.985999] kmemleak: vmalloc(64) = 000000007f7b580a
[ 8825.986025] kmemleak: kzalloc(sizeof(*elem)) = 00000000d68f3627
[ 8825.986048] kmemleak: kzalloc(sizeof(*elem)) = 000000008bcc71cd
[ 8825.986070] kmemleak: kzalloc(sizeof(*elem)) = 00000000d90adbf5
[ 8825.986092] kmemleak: kzalloc(sizeof(*elem)) = 000000004c07e127
[ 8825.986115] kmemleak: kzalloc(sizeof(*elem)) = 00000000226b752f
[ 8825.986141] kmemleak: kzalloc(sizeof(*elem)) = 00000000d7eaeed8
[ 8825.986164] kmemleak: kzalloc(sizeof(*elem)) = 000000006ed69561
[ 8825.986187] kmemleak: kzalloc(sizeof(*elem)) = 00000000a79442e4
[ 8825.986209] kmemleak: kzalloc(sizeof(*elem)) = 0000000083a42752
[ 8825.986231] kmemleak: kzalloc(sizeof(*elem)) = 00000000412c4a56
[ 8825.986259] kmemleak: kmalloc(129) = 000000005c48a002
[ 8825.986281] kmemleak: kmalloc(129) = 000000000700d3c9
[ 8825.986304] kmemleak: kmalloc(129) = 0000000000e572f9
[ 8825.986327] kmemleak: kmalloc(129) = 000000002943f11c
[ 8825.986351] kmemleak: kmalloc(129) = 00000000f9236807
[ 8825.986372] kmemleak: kmalloc(129) = 00000000b9efae8e
$ time sudo sh -c "echo scan > /sys/kernel/debug/kmemleak"

real    0m8.950s
user    0m0.000s
sys     0m8.947s
$ dmesg |tail -n1
[ 8860.390327] kmemleak: 13 new suspected memory leaks (see /sys/kernel/debug/kmemleak)
$ sudo cat /sys/kernel/debug/kmemleak
unreferenced object 0xffff88800df30540 (size 32):
  comm "modprobe", pid 5647, jiffies 4297524992 (age 866.434s)
  hex dump (first 32 bytes):
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
    00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
  backtrace:
    [<00000000c0b84cb6>] slab_post_alloc_hook+0x78/0x5b0
    [<00000000f76c1d8d>] kmem_cache_alloc_trace+0x16b/0x370
    [<00000000e1aa9887>] 0xffffffffc080f058
    [<00000000deb5ae43>] do_one_initcall+0xcb/0x430
    [<00000000fc291604>] do_init_module+0x10f/0x3b0
    [<00000000977ca321>] load_module+0x3f49/0x4570
    [<0000000040c61d85>] __do_sys_finit_module+0x12a/0x1b0
    [<00000000d87c4816>] __x64_sys_finit_module+0x43/0x50
    [<000000001a646102>] do_syscall_64+0x38/0x90
    [<0000000024b0a009>] entry_SYSCALL_64_after_hwframe+0x44/0xa9
```

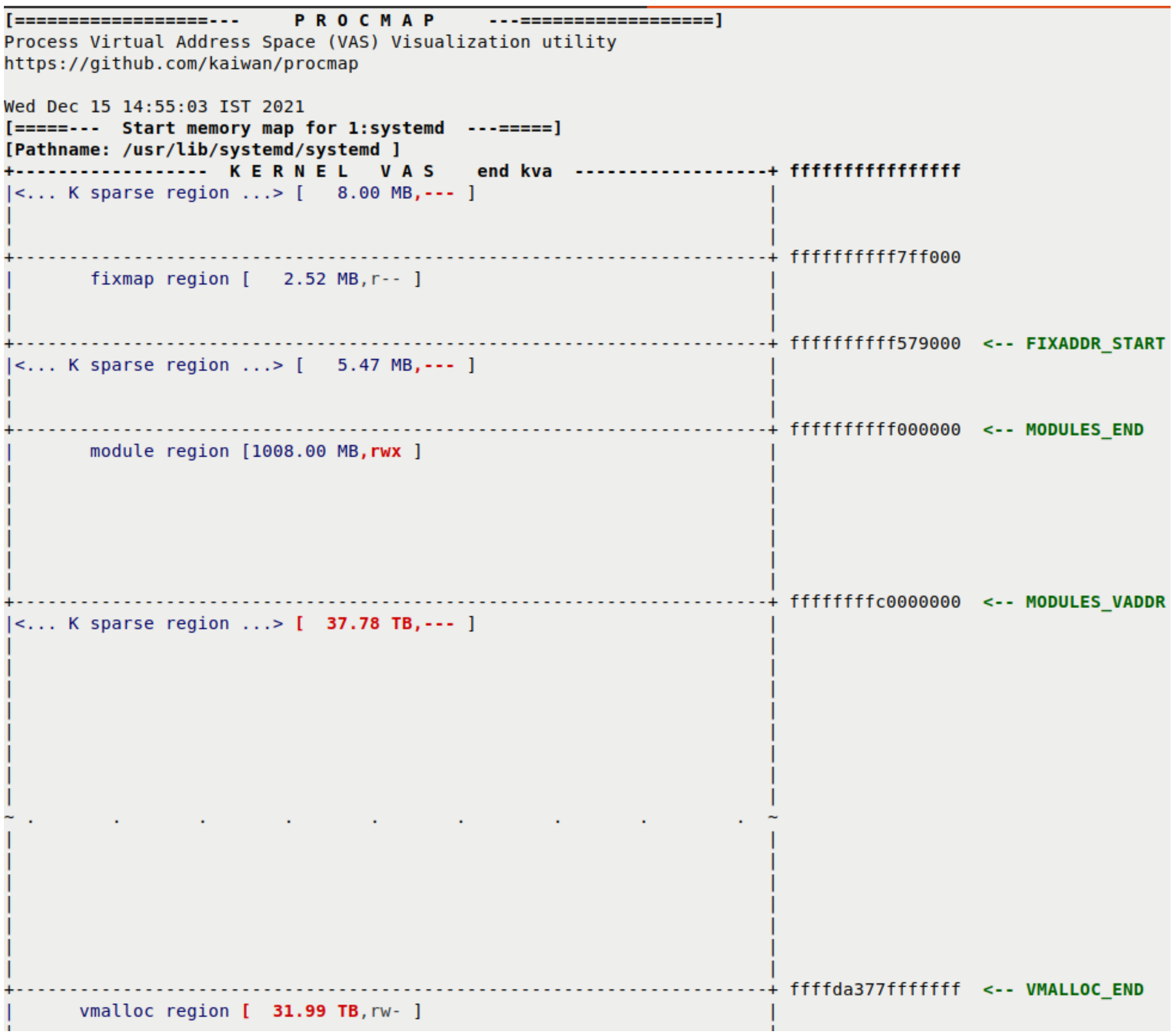
Chapter 7: Oops! Interpreting the Kernel Bug Diagnostic



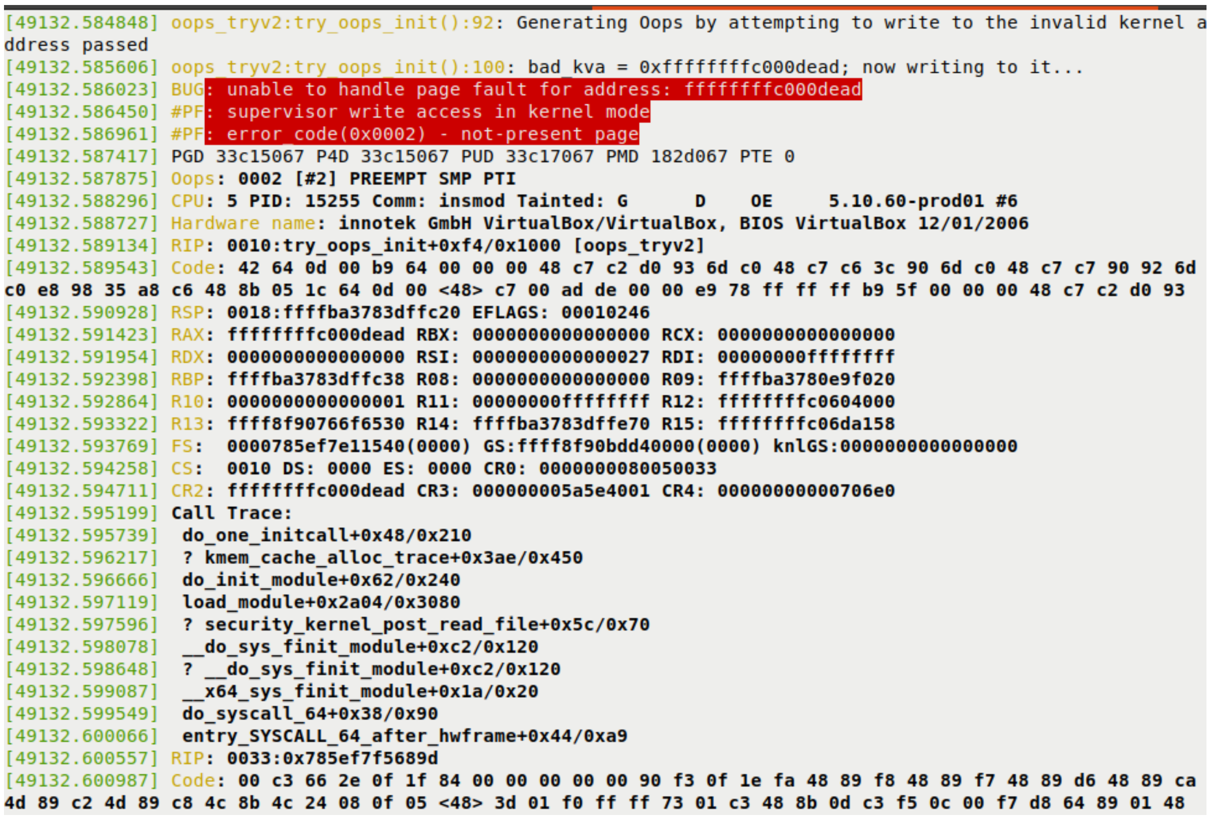
```
+------------------------------------------------------------------------+ 000057826c590000
|       /usr/bin/bash  [  36 KB,rw-,p,0x118000]                          |
+------------------------------------------------------------------------+ 000057826c587000
|       /usr/bin/bash  [  16 KB,r--,p,0x114000]                          |
+------------------------------------------------------------------------+ 000057826c583000
|       /usr/bin/bash  [ 220 KB,r--,p,0xde000]                           |
|                                                                        |
+------------------------------------------------------------------------+ 000057826c54c000
|       /usr/bin/bash  [ 708 KB,r-x,p,0x2d000]                           |
|                                                                        |
+------------------------------------------------------------------------+ 000057826c49b000
|       /usr/bin/bash  [ 180 KB,r--,p,0x0]                               |
|                                                                        |
+------------------------------------------------------------------------+ 000057826c46e000
|<... Sparse Region ...> [  87.50 TB,---,-,0x0]                          |
|                                                                        |
|                                                                        |
|                                                                        |
|                                                                        |
|                                                                        |
|                                                                        |
|                                                                        |
|                                                                        |
~ .       .       .       .       .       .        .       .        .   ~
|                                                                        |
|                                                                        |
|                                                                        |
|                                                                        |
|                                                                        |
|                                                                        |
|                                                                        |
+------------------------------------------------------------------------+ 0000000000001000
|       < NULL trap >  [   4 KB,---,-,0x0]                               |
+------------------        U S E R   V A S  start uva  -----------------+ 0000000000000000
VAS mappings:  name    [ size,perms,u:maptype,u:0xfile-offset]
```



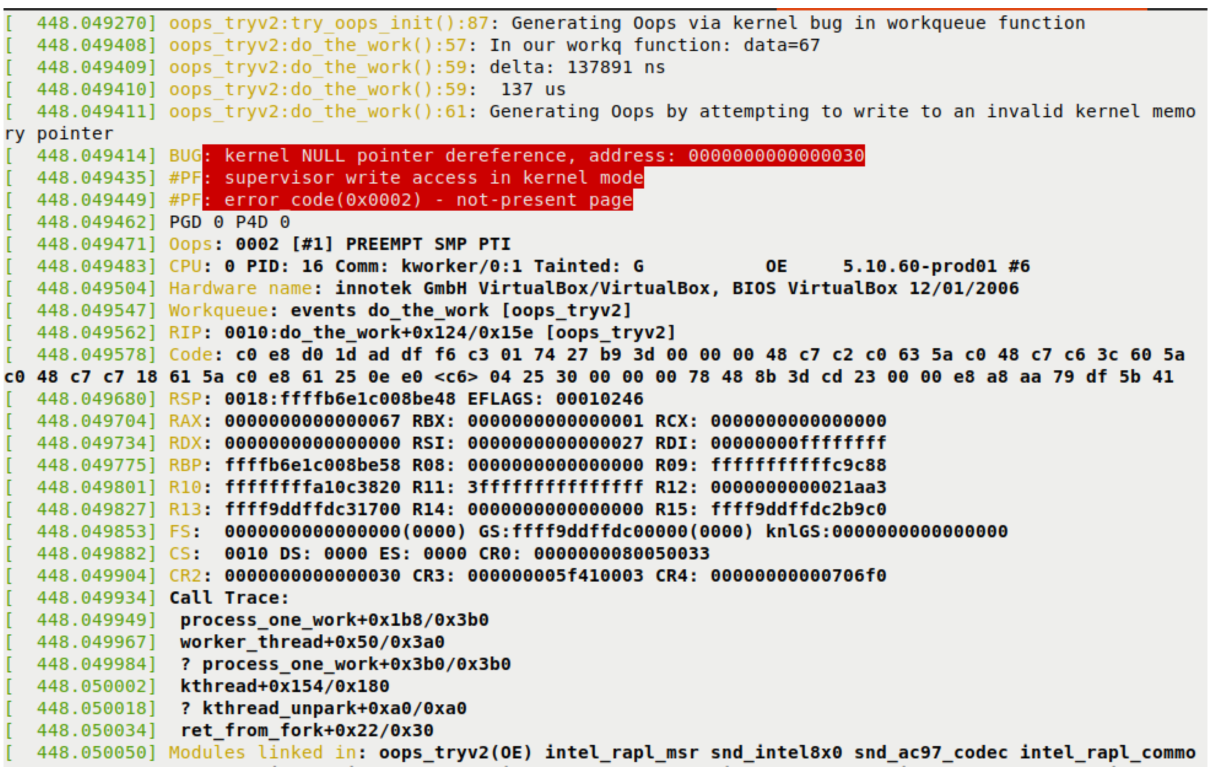
```
[  302.546331] oops_tryv1:try_oops_init():37: Lets Oops!
               Now attempting to write something to the NULL address 0x0000000000000000
[  302.546351] BUG: kernel NULL pointer dereference, address: 0000000000000000
[  302.546374] #PF: supervisor write access in kernel mode
[  302.546388] #PF: error_code(0x0002) - not-present page
[  302.546402] PGD 0 P4D 0
[  302.546411] Oops: 0002 [#1] PREEMPT SMP PTI
[  302.546424] CPU: 5 PID: 2903 Comm: insmod Tainted: G           OE     5.10.60-prod01 #6
[  302.546466] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[  302.546489] RIP: 0010:try_oops_init+0xdb/0x1000 [oops_tryv1]
```



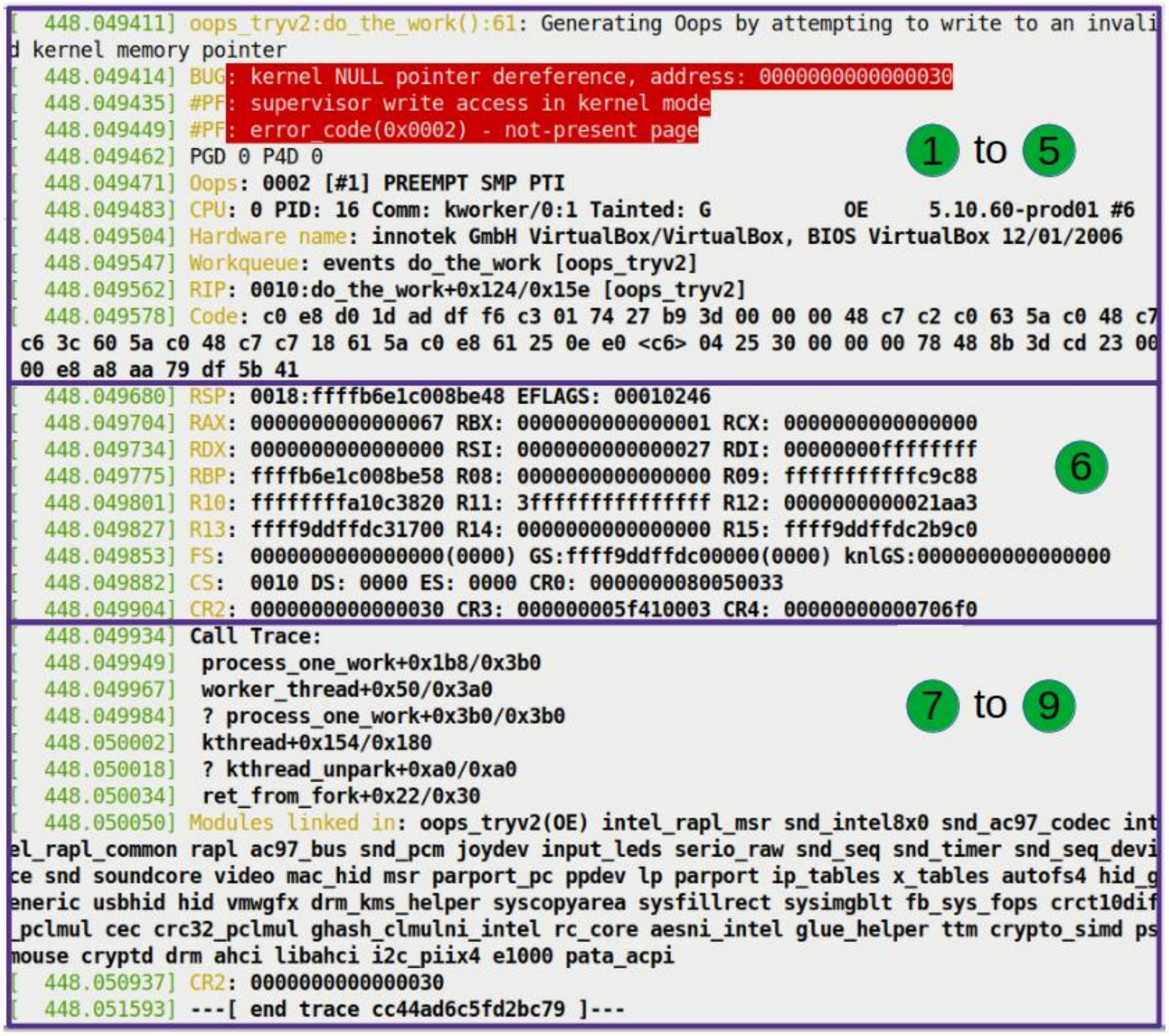
```
[==================---     P R O C M A P     ---==================]
Process Virtual Address Space (VAS) Visualization utility
https://github.com/kaiwan/procmap

Wed Dec 15 14:55:03 IST 2021
[=====---  Start memory map for 1:systemd  ---=====]
[Pathname: /usr/lib/systemd/systemd ]
+------------------  K E R N E L   V A S    end kva  ------------------+ ffffffffffffffff
|<... K sparse region ...> [   8.00 MB,--- ]                           |
|                                                                      |
|                                                                      |
+----------------------------------------------------------------------+ ffffffffff7ff000
|       fixmap region [   2.52 MB,r-- ]                                |
|                                                                      |
|                                                                      |
+----------------------------------------------------------------------+ ffffffffff579000  <-- FIXADDR_START
|<... K sparse region ...> [   5.47 MB,--- ]                           |
|                                                                      |
|                                                                      |
+----------------------------------------------------------------------+ ffffffffff000000  <-- MODULES_END
|       module region [1008.00 MB,rwx ]                                |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
+----------------------------------------------------------------------+ ffffffffc0000000  <-- MODULES_VADDR
|<... K sparse region ...> [  37.78 TB,--- ]                           |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
~ .       .       .       .       .       .        .       .        .  ~
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
+----------------------------------------------------------------------+ ffffda377fffffff  <-- VMALLOC_END
|      vmalloc region [  31.99 TB,rw- ]                                |
```



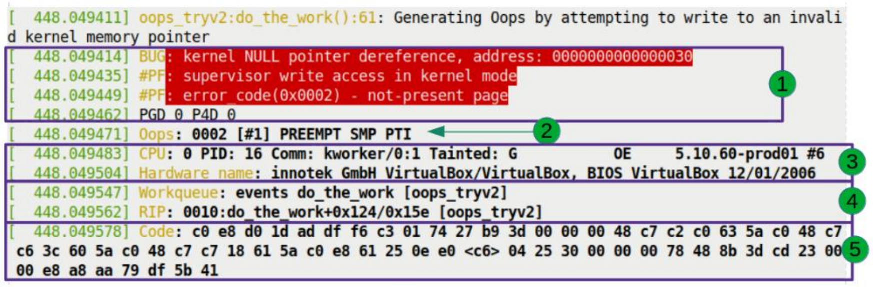
```
[49132.584848] oops_tryv2:try_oops_init():92: Generating Oops by attempting to write to the invalid kernel a
ddress passed
[49132.585606] oops_tryv2:try_oops_init():100: bad_kva = 0xffffffffc000dead; now writing to it...
[49132.586023] BUG: unable to handle page fault for address: ffffffffc000dead
[49132.586450] #PF: supervisor write access in kernel mode
[49132.586961] #PF: error_code(0x0002) - not-present page
[49132.587417] PGD 33c15067 P4D 33c15067 PUD 33c17067 PMD 182d067 PTE 0
[49132.587875] Oops: 0002 [#2] PREEMPT SMP PTI
[49132.588296] CPU: 5 PID: 15255 Comm: insmod Tainted: G      D    OE     5.10.60-prod01 #6
[49132.588727] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[49132.589134] RIP: 0010:try_oops_init+0xf4/0x1000 [oops_tryv2]
[49132.589543] Code: 42 64 0d 00 b9 64 00 00 00 48 c7 c2 d0 93 6d c0 48 c7 c6 3c 90 6d c0 48 c7 c7 90 92 6d
c0 e8 98 35 a8 c6 48 8b 05 1c 64 0d 00 <48> c7 00 ad de 00 00 e9 78 ff ff ff b9 5f 00 00 00 48 c7 c2 d0 93
[49132.590928] RSP: 0018:ffffba3783dffc20 EFLAGS: 00010246
[49132.591423] RAX: ffffffffc000dead RBX: 0000000000000000 RCX: 0000000000000000
[49132.591954] RDX: 0000000000000000 RSI: 0000000000000027 RDI: 00000000ffffffff
[49132.592398] RBP: ffffba3783dffc38 R08: 0000000000000000 R09: ffffba3780e9f020
[49132.592864] R10: 0000000000000001 R11: 00000000ffffffff R12: ffffffffc0604000
[49132.593322] R13: ffff8f90766f6530 R14: ffffba3783dffe70 R15: ffffffffc06da158
[49132.593769] FS:  0000785ef7e11540(0000) GS:ffff8f90bdd40000(0000) knlGS:0000000000000000
[49132.594258] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[49132.594711] CR2: ffffffffc000dead CR3: 000000005a5e4001 CR4: 00000000000706e0
[49132.595199] Call Trace:
[49132.595739]  do_one_initcall+0x48/0x210
[49132.596217]  ? kmem_cache_alloc_trace+0x3ae/0x450
[49132.596666]  do_init_module+0x62/0x240
[49132.597119]  load_module+0x2a04/0x3080
[49132.597596]  ? security_kernel_post_read_file+0x5c/0x70
[49132.598078]  __do_sys_finit_module+0xc2/0x120
[49132.598648]  ? __do_sys_finit_module+0xc2/0x120
[49132.599087]  __x64_sys_finit_module+0x1a/0x20
[49132.599549]  do_syscall_64+0x38/0x90
[49132.600066]  entry_SYSCALL_64_after_hwframe+0x44/0xa9
[49132.600557] RIP: 0033:0x785ef7f5689d
[49132.600987] Code: 00 c3 66 2e 0f 1f 84 00 00 00 00 00 90 f3 0f 1e fa 48 89 f8 48 89 f7 48 89 d6 48 89 ca
4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d c3 f5 0c 00 f7 d8 64 89 01 48
```



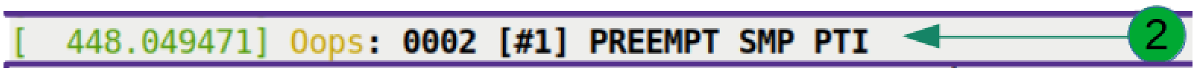
```
[  448.049270] oops_tryv2:try_oops_init():87: Generating Oops via kernel bug in workqueue function
[  448.049408] oops_tryv2:do_the_work():57: In our workq function: data=67
[  448.049409] oops_tryv2:do_the_work():59: delta: 137891 ns
[  448.049410] oops_tryv2:do_the_work():59:  137 us
[  448.049411] oops_tryv2:do_the_work():61: Generating Oops by attempting to write to an invalid kernel memo
ry pointer
[  448.049414] BUG: kernel NULL pointer dereference, address: 0000000000000030
[  448.049435] #PF: supervisor write access in kernel mode
[  448.049449] #PF: error_code(0x0002) - not-present page
[  448.049462] PGD 0 P4D 0
[  448.049471] Oops: 0002 [#1] PREEMPT SMP PTI
[  448.049483] CPU: 0 PID: 16 Comm: kworker/0:1 Tainted: G           OE     5.10.60-prod01 #6
[  448.049504] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[  448.049547] Workqueue: events do_the_work [oops_tryv2]
[  448.049562] RIP: 0010:do_the_work+0x124/0x15e [oops_tryv2]
[  448.049578] Code: c0 e8 d0 1d ad df f6 c3 01 74 27 b9 3d 00 00 00 48 c7 c2 c0 63 5a c0 48 c7 c6 3c 60 5a
c0 48 c7 c7 18 61 5a c0 e8 61 25 0e e0 <c6> 04 25 30 00 00 00 78 48 8b 3d cd 23 00 00 e8 a8 aa 79 df 5b 41
[  448.049680] RSP: 0018:ffffb6e1c008be48 EFLAGS: 00010246
[  448.049704] RAX: 0000000000000067 RBX: 0000000000000001 RCX: 0000000000000000
[  448.049734] RDX: 0000000000000000 RSI: 0000000000000027 RDI: 00000000ffffffff
[  448.049775] RBP: ffffb6e1c008be58 R08: 0000000000000000 R09: fffffffffffc9c88
[  448.049801] R10: ffffffffa10c3820 R11: 3fffffffffffffff R12: 0000000000021aa3
[  448.049827] R13: ffff9ddffdc31700 R14: 0000000000000000 R15: ffff9ddffdc2b9c0
[  448.049853] FS:  0000000000000000(0000) GS:ffff9ddffdc00000(0000) knlGS:0000000000000000
[  448.049882] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[  448.049904] CR2: 0000000000000030 CR3: 000000005f410003 CR4: 00000000000706f0
[  448.049934] Call Trace:
[  448.049949]  process_one_work+0x1b8/0x3b0
[  448.049967]  worker_thread+0x50/0x3a0
[  448.049984]  ? process_one_work+0x3b0/0x3b0
[  448.050002]  kthread+0x154/0x180
[  448.050018]  ? kthread_unpark+0xa0/0xa0
[  448.050034]  ret_from_fork+0x22/0x30
[  448.050050] Modules linked in: oops_tryv2(OE) intel_rapl_msr snd_intel8x0 snd_ac97_codec intel_rapl_commo
```



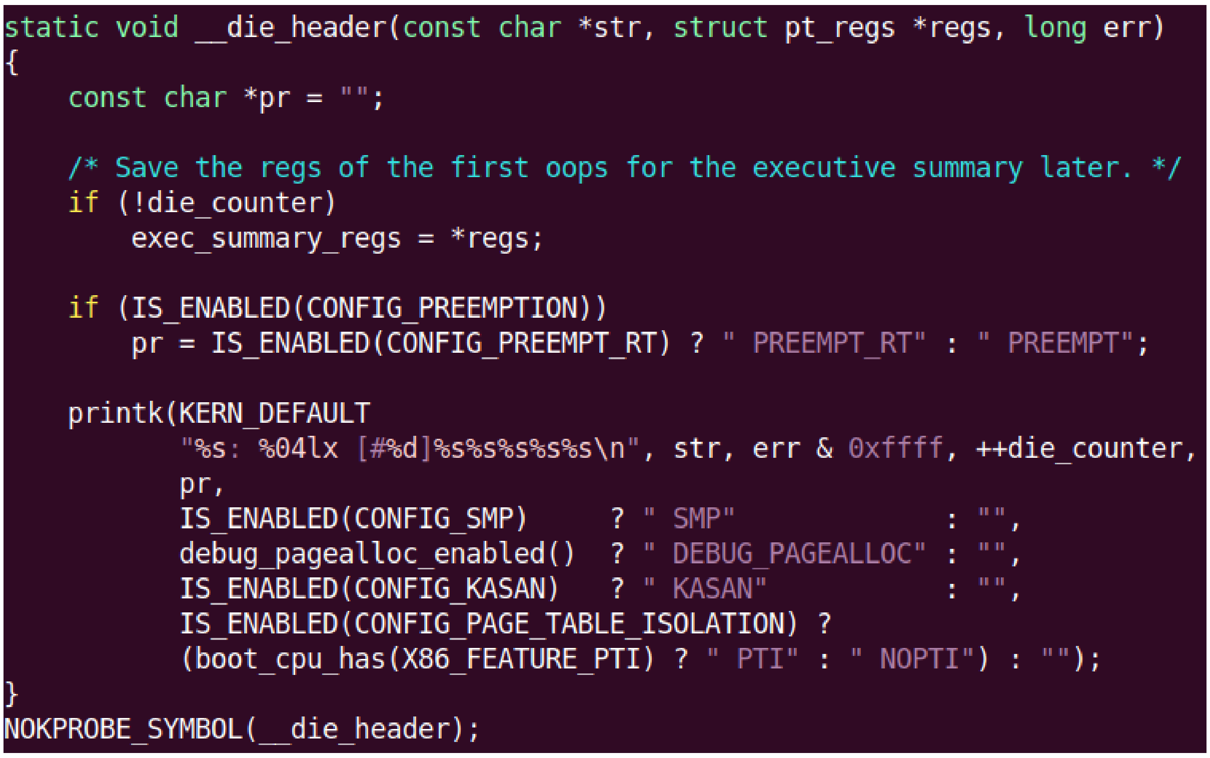
```
[  448.049411] oops_tryv2:do_the_work():61: Generating Oops by attempting to write to an invali
d kernel memory pointer
[  448.049414] BUG: kernel NULL pointer dereference, address: 0000000000000030
[  448.049435] #PF: supervisor write access in kernel mode
[  448.049449] #PF: error_code(0x0002) - not-present page
[  448.049462] PGD 0 P4D 0
[  448.049471] Oops: 0002 [#1] PREEMPT SMP PTI
[  448.049483] CPU: 0 PID: 16 Comm: kworker/0:1 Tainted: G           OE     5.10.60-prod01 #6
[  448.049504] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[  448.049547] Workqueue: events do_the_work [oops_tryv2]
[  448.049562] RIP: 0010:do_the_work+0x124/0x15e [oops_tryv2]
[  448.049578] Code: c0 e8 d0 1d ad df f6 c3 01 74 27 b9 3d 00 00 00 48 c7 c2 c0 63 5a c0 48 c7
 c6 3c 60 5a c0 48 c7 c7 18 61 5a c0 e8 61 25 0e e0 <c6> 04 25 30 00 00 00 78 48 8b 3d cd 23 00
 00 e8 a8 aa 79 df 5b 41
[  448.049680] RSP: 0018:ffffb6e1c008be48 EFLAGS: 00010246
[  448.049704] RAX: 0000000000000067 RBX: 0000000000000001 RCX: 0000000000000000
[  448.049734] RDX: 0000000000000000 RSI: 0000000000000027 RDI: 00000000ffffffff
[  448.049775] RBP: ffffb6e1c008be58 R08: 0000000000000000 R09: ffffffffffc9c88
[  448.049801] R10: ffffffffa10c3820 R11: 3fffffffffffffff R12: 0000000000021aa3
[  448.049827] R13: ffff9ddffdc31700 R14: 0000000000000000 R15: ffff9ddffdc2b9c0
[  448.049853] FS:  0000000000000000(0000) GS:ffff9ddffdc00000(0000) knlGS:0000000000000000
[  448.049882] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[  448.049904] CR2: 0000000000000030 CR3: 000000005f410003 CR4: 00000000000706f0
[  448.049934] Call Trace:
[  448.049949]  process_one_work+0x1b8/0x3b0
[  448.049967]  worker_thread+0x50/0x3a0
[  448.049984]  ? process_one_work+0x3b0/0x3b0
[  448.050002]  kthread+0x154/0x180
[  448.050018]  ? kthread_unpark+0xa0/0xa0
[  448.050034]  ret_from_fork+0x22/0x30
[  448.050050] Modules linked in: oops_tryv2(OE) intel_rapl_msr snd_intel8x0 snd_ac97_codec int
el_rapl_common rapl ac97_bus snd_pcm joydev input_leds serio_raw snd_seq snd_timer snd_seq_devi
ce snd soundcore video mac_hid msr parport_pc ppdev lp parport ip_tables x_tables autofs4 hid_g
eneric usbhid hid vmwgfx drm_kms_helper syscopyarea sysfillrect sysimgblt fb_sys_fops crct10dif
_pclmul cec crc32_pclmul ghash_clmulni_intel rc_core aesni_intel glue_helper ttm crypto_simd ps
mouse cryptd drm ahci libahci i2c_piix4 e1000 pata_acpi
[  448.050937] CR2: 0000000000000030
[  448.051593] ---[ end trace cc44ad6c5fd2bc79 ]---
```



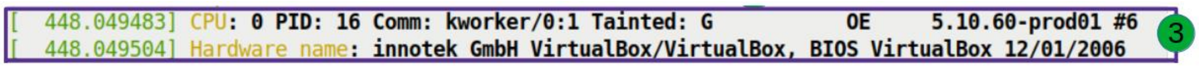
```
[  448.049411] oops_tryv2:do_the_work():61: Generating Oops by attempting to write to an invali
d kernel memory pointer
[  448.049414] BUG: kernel NULL pointer dereference, address: 0000000000000030
[  448.049435] #PF: supervisor write access in kernel mode
[  448.049449] #PF: error_code(0x0002) - not-present page
[  448.049462] PGD 0 P4D 0
[  448.049471] Oops: 0002 [#1] PREEMPT SMP PTI
[  448.049483] CPU: 0 PID: 16 Comm: kworker/0:1 Tainted: G           OE     5.10.60-prod01 #6
[  448.049504] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[  448.049547] Workqueue: events do_the_work [oops_tryv2]
[  448.049562] RIP: 0010:do_the_work+0x124/0x15e [oops_tryv2]
[  448.049578] Code: c0 e8 d0 1d ad df f6 c3 01 74 27 b9 3d 00 00 00 48 c7 c2 c0 63 5a c0 48 c7
 c6 3c 60 5a c0 48 c7 c7 18 61 5a c0 e8 61 25 0e e0 <c6> 04 25 30 00 00 00 78 48 8b 3d cd 23 00
 00 e8 a8 aa 79 df 5b 41
```



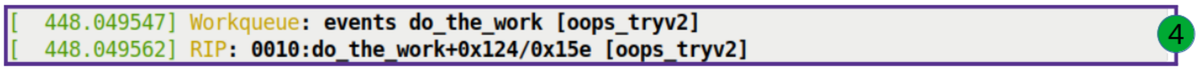
```
[  448.049471] Oops: 0002 [#1] PREEMPT SMP PTI
```



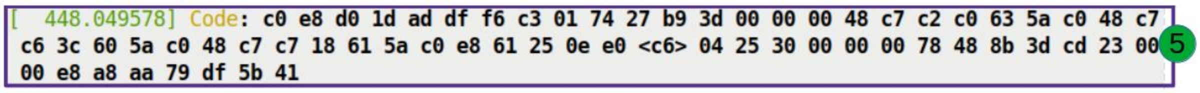
```
static void __die_header(const char *str, struct pt_regs *regs, long err)
{
    const char *pr = "";

    /* Save the regs of the first oops for the executive summary later. */
    if (!die_counter)
        exec_summary_regs = *regs;

    if (IS_ENABLED(CONFIG_PREEMPTION))
        pr = IS_ENABLED(CONFIG_PREEMPT_RT) ? " PREEMPT_RT" : " PREEMPT";

    printk(KERN_DEFAULT
           "%s: %04lx [#%d]%s%s%s%s%s\n", str, err & 0xffff, ++die_counter,
           pr,
           IS_ENABLED(CONFIG_SMP)      ? " SMP"             : "",
           debug_pagealloc_enabled()   ? " DEBUG_PAGEALLOC" : "",
           IS_ENABLED(CONFIG_KASAN)    ? " KASAN"           : "",
           IS_ENABLED(CONFIG_PAGE_TABLE_ISOLATION) ?
           (boot_cpu_has(X86_FEATURE_PTI) ? " PTI" : " NOPTI") : "");
}
NOKPROBE_SYMBOL(__die_header);
```



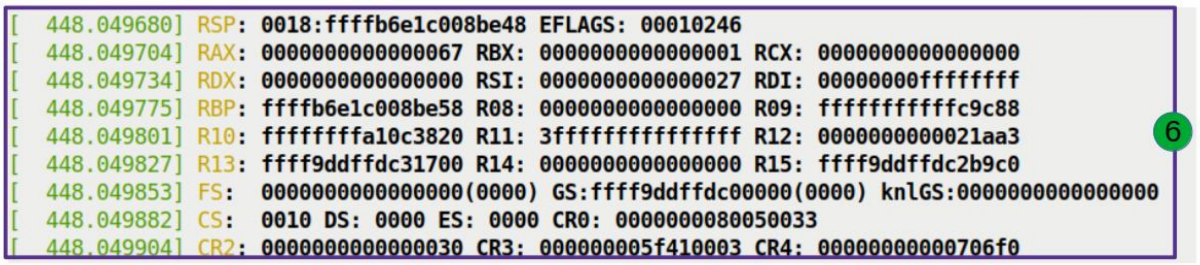
```
[  448.049483] CPU: 0 PID: 16 Comm: kworker/0:1 Tainted: G           OE     5.10.60-prod01 #6
[  448.049504] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
```



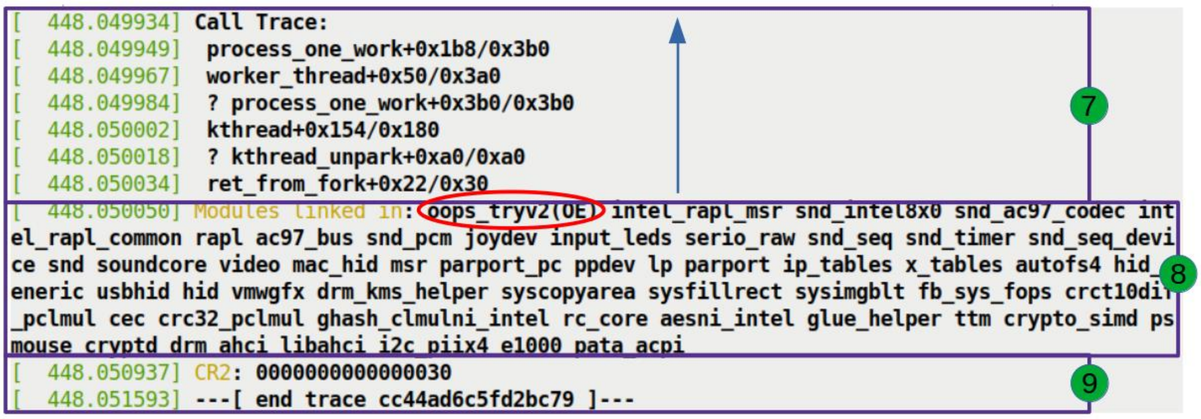
```
[  448.049547] Workqueue: events do_the_work [oops_tryv2]
[  448.049562] RIP: 0010:do_the_work+0x124/0x15e [oops_tryv2]
```



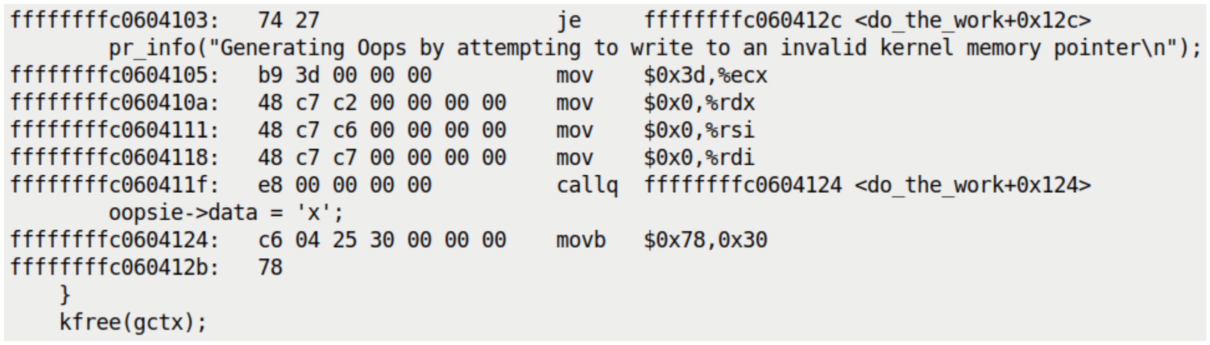
```
[  448.049578] Code: c0 e8 d0 1d ad df f6 c3 01 74 27 b9 3d 00 00 00 48 c7 c2 c0 63 5a c0 48 c7
 c6 3c 60 5a c0 48 c7 c7 18 61 5a c0 e8 61 25 0e e0 <c6> 04 25 30 00 00 00 78 48 8b 3d cd 23 00
 00 e8 a8 aa 79 df 5b 41
```



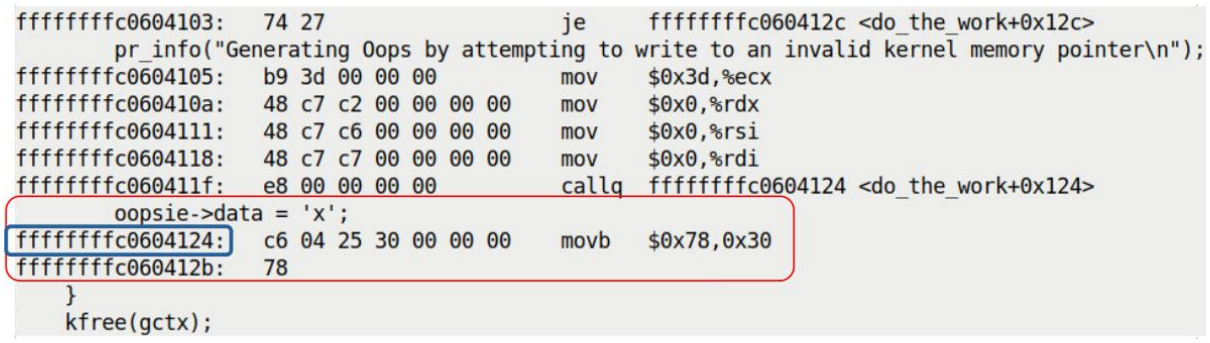
```
[  448.049680] RSP: 0018:ffffb6e1c008be48 EFLAGS: 00010246
[  448.049704] RAX: 0000000000000067 RBX: 0000000000000001 RCX: 0000000000000000
[  448.049734] RDX: 0000000000000000 RSI: 0000000000000027 RDI: 00000000ffffffff
[  448.049775] RBP: ffffb6e1c008be58 R08: 0000000000000000 R09: ffffffffffffc9c88
[  448.049801] R10: ffffffffa10c3820 R11: 3fffffffffffffff R12: 000000000021aa3
[  448.049827] R13: ffff9ddffdc31700 R14: 0000000000000000 R15: ffff9ddffdc2b9c0
[  448.049853] FS:  0000000000000000(0000) GS:ffff9ddffdc00000(0000) knlGS:0000000000000000
[  448.049882] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[  448.049904] CR2: 0000000000000030 CR3: 000000005f410003 CR4: 00000000000706f0
```



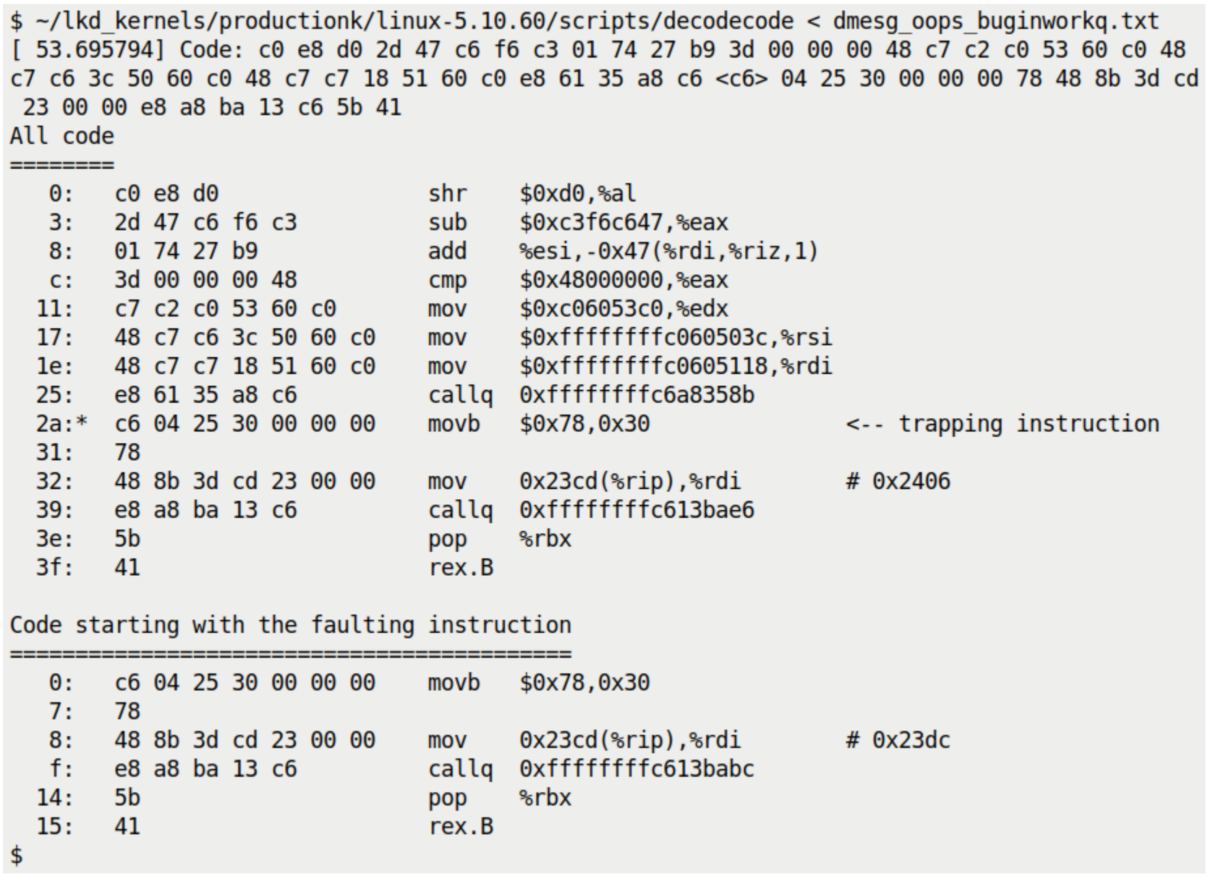
```
[  448.049934] Call Trace:
[  448.049949]  process_one_work+0x1b8/0x3b0
[  448.049967]  worker_thread+0x50/0x3a0
[  448.049984]  ? process_one_work+0x3b0/0x3b0
[  448.050002]  kthread+0x154/0x180
[  448.050018]  ? kthread_unpark+0xa0/0xa0
[  448.050034]  ret_from_fork+0x22/0x30
[  448.050050] Modules linked in: oops_tryv2(OE) intel_rapl_msr snd_intel8x0 snd_ac97_codec int
el_rapl_common rapl ac97_bus snd_pcm joydev input_leds serio_raw snd_seq snd_timer snd_seq_devi
ce snd soundcore video mac_hid msr parport_pc ppdev lp parport ip_tables x_tables autofs4 hid_g
eneric usbhid hid vmwgfx drm_kms_helper syscopyarea sysfillrect sysimgblt fb_sys_fops crct10dif
_pclmul cec crc32_pclmul ghash_clmulni_intel rc_core aesni_intel glue_helper ttm crypto_simd ps
mouse cryptd drm ahci libahci i2c_piix4 e1000 pata_acpi
[  448.050937] CR2: 0000000000000030
[  448.051593] ---[ end trace cc44ad6c5fd2bc79 ]---
```



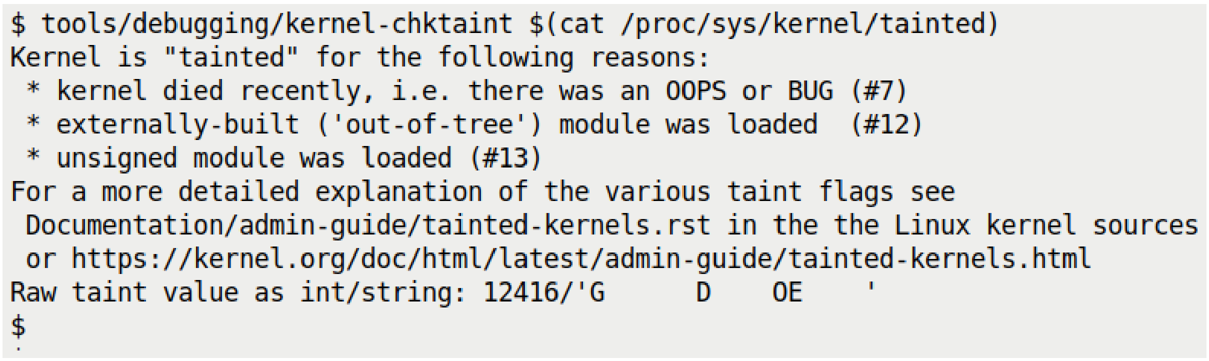
```
ffffffffc0604103:   74 27                   je     ffffffffc060412c <do_the_work+0x12c>
        pr_info("Generating Oops by attempting to write to an invalid kernel memory pointer\n");
ffffffffc0604105:   b9 3d 00 00 00          mov    $0x3d,%ecx
ffffffffc060410a:   48 c7 c2 00 00 00 00    mov    $0x0,%rdx
ffffffffc0604111:   48 c7 c6 00 00 00 00    mov    $0x0,%rsi
ffffffffc0604118:   48 c7 c7 00 00 00 00    mov    $0x0,%rdi
ffffffffc060411f:   e8 00 00 00 00          callq  ffffffffc0604124 <do_the_work+0x124>
        oopsie->data = 'x';
ffffffffc0604124:   c6 04 25 30 00 00 00    movb   $0x78,0x30
ffffffffc060412b:   78
    }
    kfree(gctx);
```



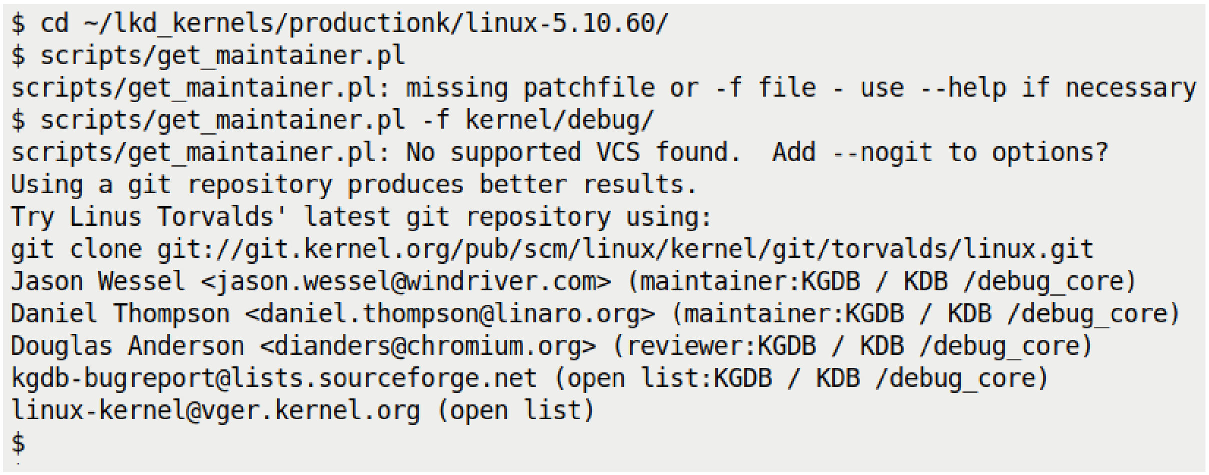
```
ffffffffc0604103:   74 27                   je     ffffffffc060412c <do_the_work+0x12c>
        pr_info("Generating Oops by attempting to write to an invalid kernel memory pointer\n");
ffffffffc0604105:   b9 3d 00 00 00          mov    $0x3d,%ecx
ffffffffc060410a:   48 c7 c2 00 00 00 00    mov    $0x0,%rdx
ffffffffc0604111:   48 c7 c6 00 00 00 00    mov    $0x0,%rsi
ffffffffc0604118:   48 c7 c7 00 00 00 00    mov    $0x0,%rdi
ffffffffc060411f:   e8 00 00 00 00          callq  ffffffffc0604124 <do_the_work+0x124>
        oopsie->data = 'x';
ffffffffc0604124:   c6 04 25 30 00 00 00    movb   $0x78,0x30
ffffffffc060412b:   78
    }
    kfree(gctx);
```



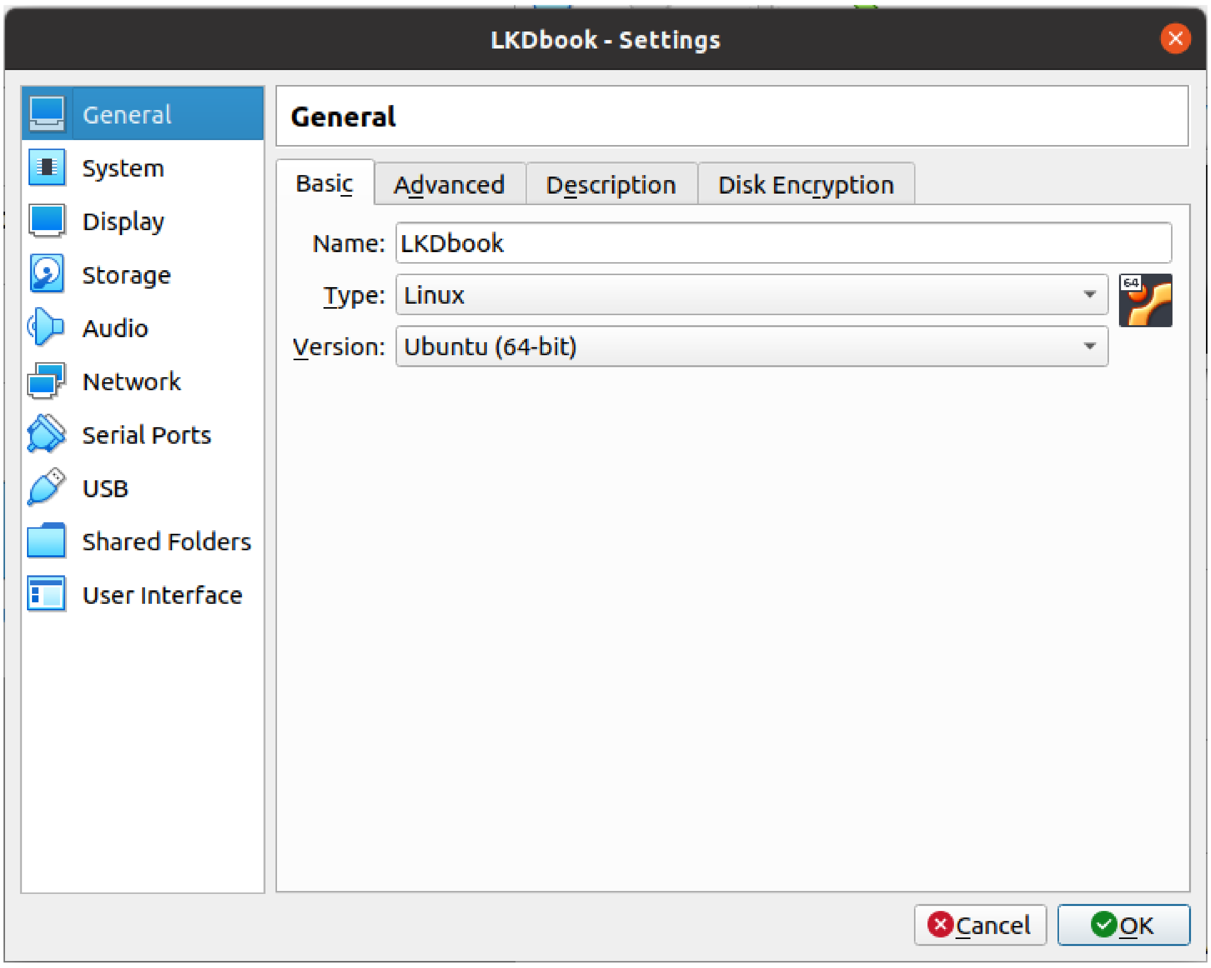
```
$ ~/lkd_kernels/productionk/linux-5.10.60/scripts/decodecode < dmesg_oops_buginworkq.txt
[ 53.695794] Code: c0 e8 d0 2d 47 c6 f6 c3 01 74 27 b9 3d 00 00 00 48 c7 c2 c0 53 60 c0 48
c7 c6 3c 50 60 c0 48 c7 c7 18 51 60 c0 e8 61 35 a8 c6 <c6> 04 25 30 00 00 00 78 48 8b 3d cd
 23 00 00 e8 a8 ba 13 c6 5b 41
All code
========
   0:   c0 e8 d0                shr    $0xd0,%al
   3:   2d 47 c6 f6 c3          sub    $0xc3f6c647,%eax
   8:   01 74 27 b9             add    %esi,-0x47(%rdi,%riz,1)
   c:   3d 00 00 00 48          cmp    $0x48000000,%eax
  11:   c7 c2 c0 53 60 c0       mov    $0xc06053c0,%edx
  17:   48 c7 c6 3c 50 60 c0    mov    $0xffffffffc060503c,%rsi
  1e:   48 c7 c7 18 51 60 c0    mov    $0xffffffffc0605118,%rdi
  25:   e8 61 35 a8 c6          callq  0xffffffffc6a8358b
  2a:*  c6 04 25 30 00 00 00    movb   $0x78,0x30               <-- trapping instruction
  31:   78
  32:   48 8b 3d cd 23 00 00    mov    0x23cd(%rip),%rdi        # 0x2406
  39:   e8 a8 ba 13 c6          callq  0xffffffffc613bae6
  3e:   5b                      pop    %rbx
  3f:   41                      rex.B

Code starting with the faulting instruction
===========================================
   0:   c6 04 25 30 00 00 00    movb   $0x78,0x30
   7:   78
   8:   48 8b 3d cd 23 00 00    mov    0x23cd(%rip),%rdi        # 0x23dc
   f:   e8 a8 ba 13 c6          callq  0xffffffffc613babc
  14:   5b                      pop    %rbx
  15:   41                      rex.B
$
```



```
$ tools/debugging/kernel-chktaint $(cat /proc/sys/kernel/tainted)
Kernel is "tainted" for the following reasons:
 * kernel died recently, i.e. there was an OOPS or BUG (#7)
 * externally-built ('out-of-tree') module was loaded  (#12)
 * unsigned module was loaded (#13)
For a more detailed explanation of the various taint flags see
 Documentation/admin-guide/tainted-kernels.rst in the the Linux kernel sources
 or https://kernel.org/doc/html/latest/admin-guide/tainted-kernels.html
Raw taint value as int/string: 12416/'G      D    OE    '
$
```



```
$ cd ~/lkd_kernels/productionk/linux-5.10.60/
$ scripts/get_maintainer.pl
scripts/get_maintainer.pl: missing patchfile or -f file - use --help if necessary
$ scripts/get_maintainer.pl -f kernel/debug/
scripts/get_maintainer.pl: No supported VCS found.  Add --nogit to options?
Using a git repository produces better results.
Try Linus Torvalds' latest git repository using:
git clone git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git
Jason Wessel <jason.wessel@windriver.com> (maintainer:KGDB / KDB /debug_core)
Daniel Thompson <daniel.thompson@linaro.org> (maintainer:KGDB / KDB /debug_core)
Douglas Anderson <dianders@chromium.org> (reviewer:KGDB / KDB /debug_core)
kgdb-bugreport@lists.sourceforge.net (open list:KGDB / KDB /debug_core)
linux-kernel@vger.kernel.org (open list)
$
```



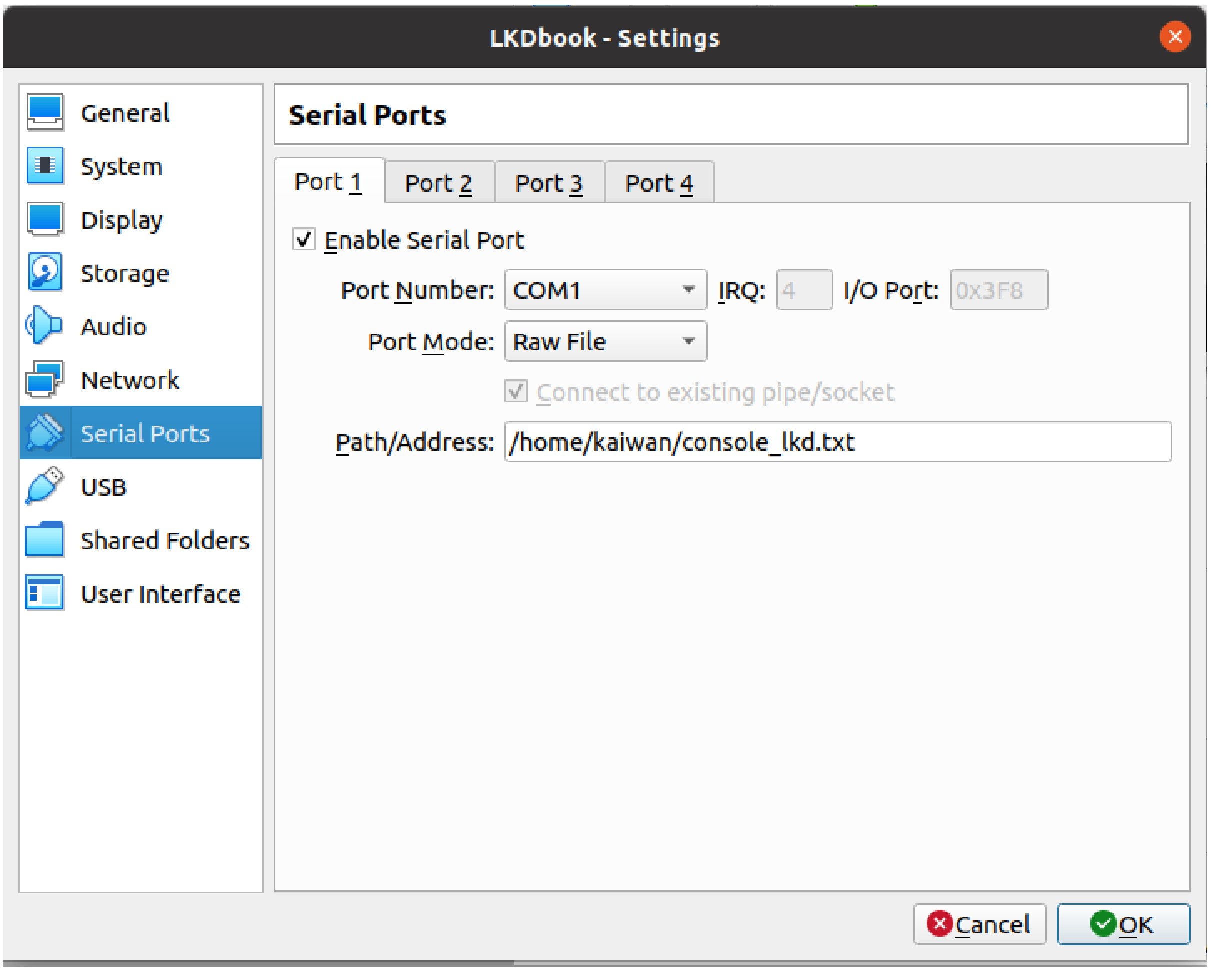
LKDbook - Settings

General

Name: LKDbook

Type: Linux

Version: Ubuntu (64-bit)



Serial Ports

Port 1 | Port 2 | Port 3 | Port 4

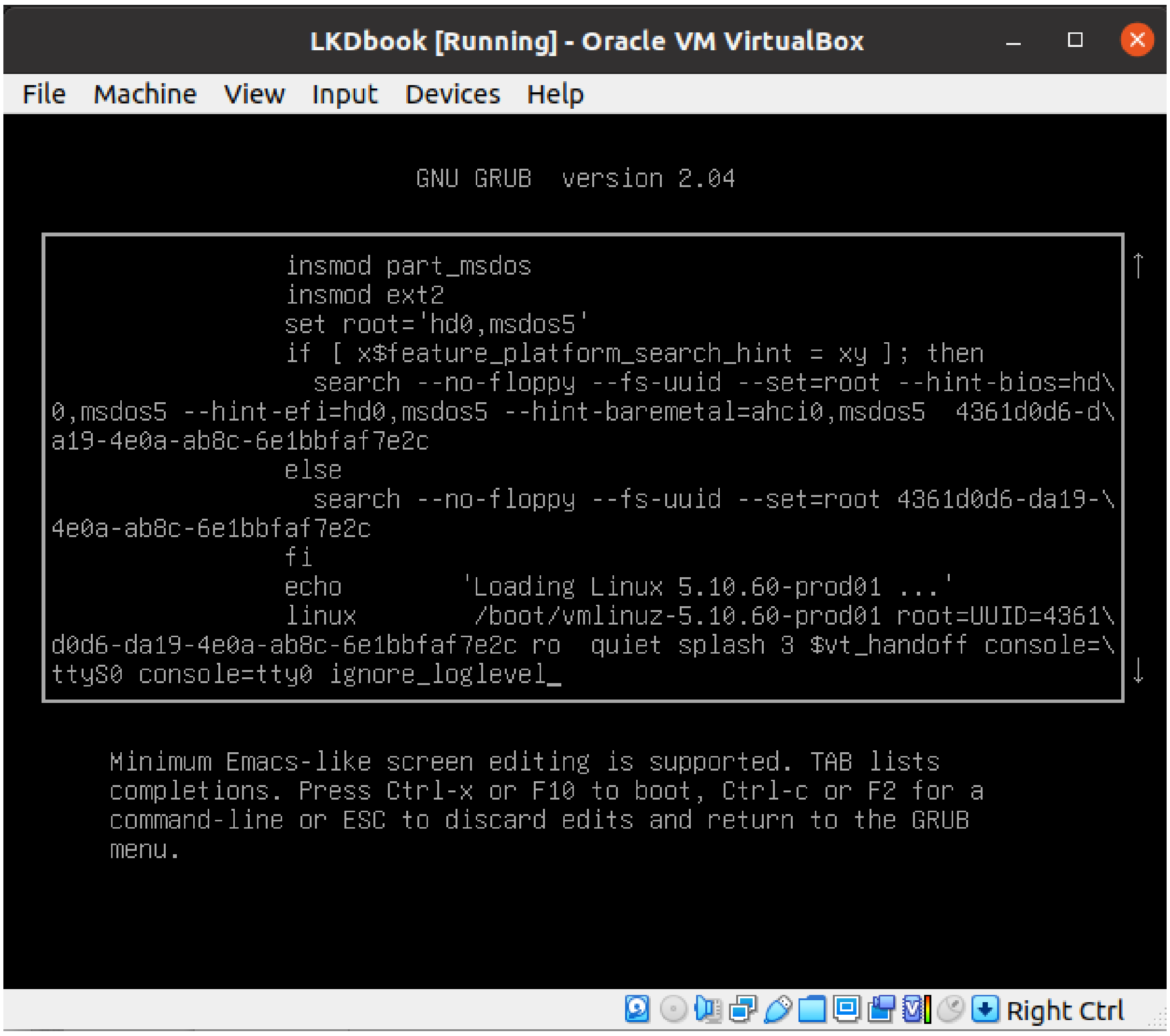
[✓] Enable Serial Port

Port Number: COM1 IRQ: 4 I/O Port: 0x3F8

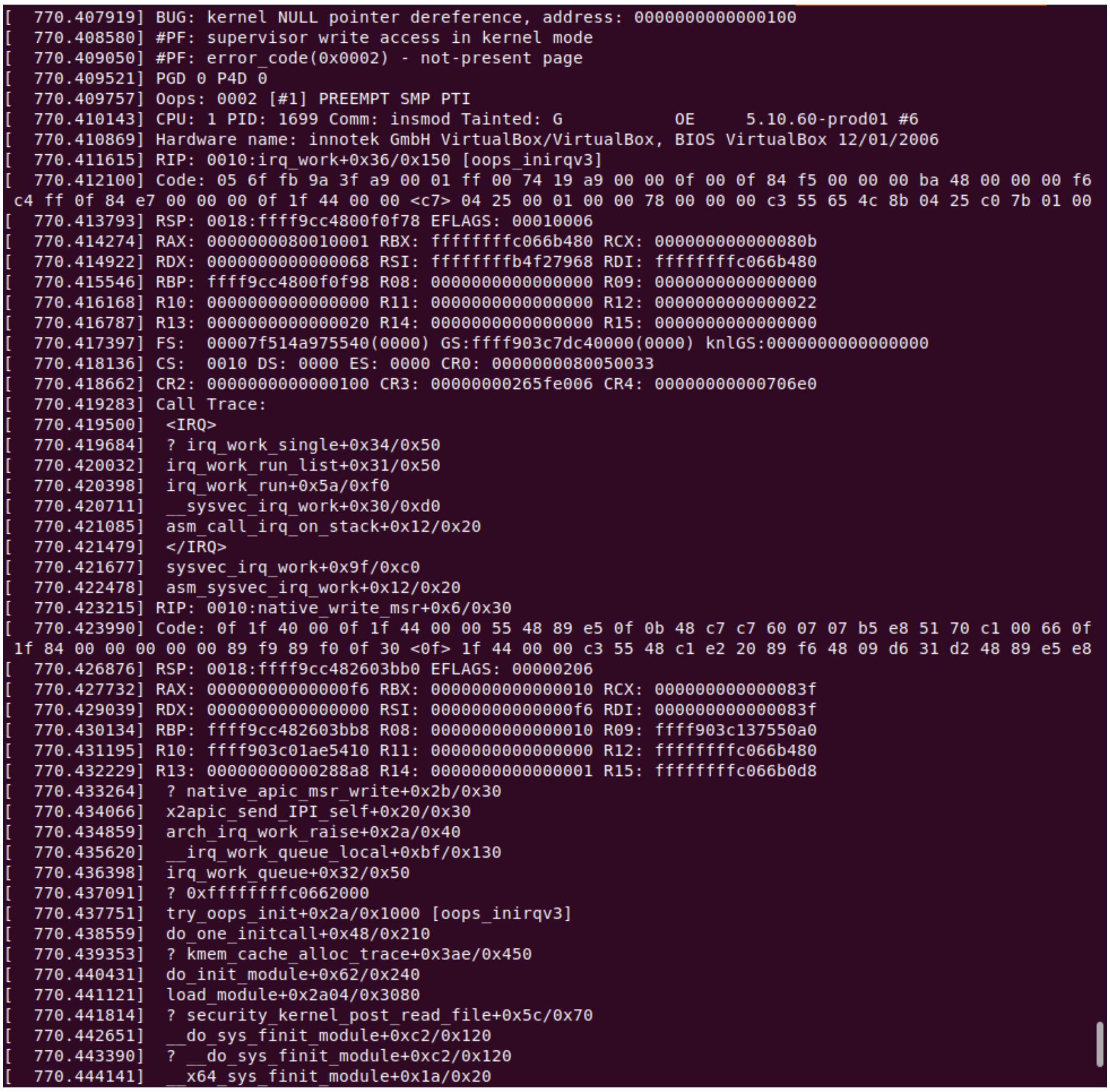
Port Mode: Raw File

[✓] Connect to existing pipe/socket

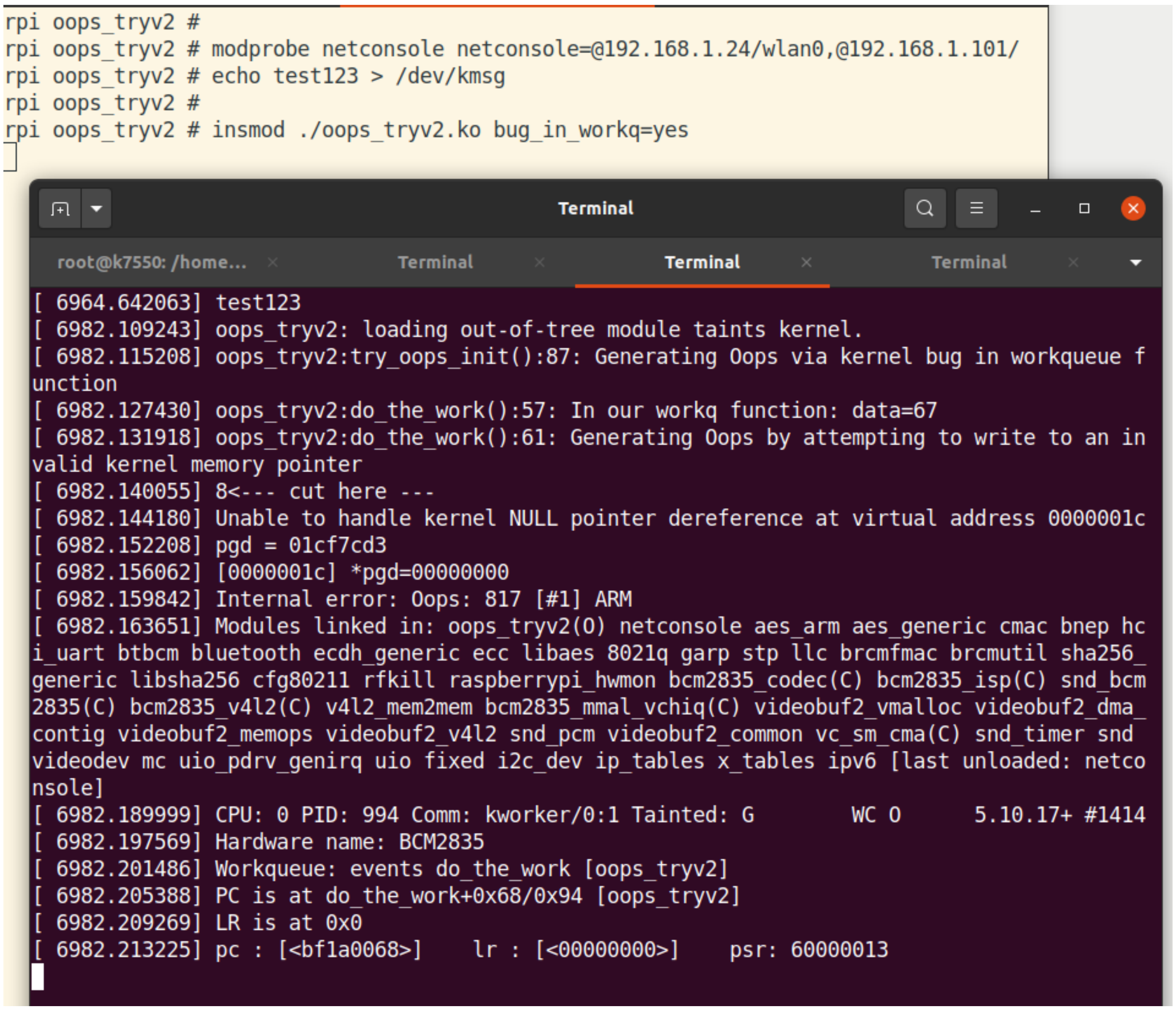
Path/Address: /home/kaiwan/console_lkd.txt


LKDbook [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
                     GNU GRUB  version 2.04

                insmod part_msdos
                insmod ext2
                set root='hd0,msdos5'
                if [ x$feature_platform_search_hint = xy ]; then
                  search --no-floppy --fs-uuid --set=root --hint-bios=hd\
0,msdos5 --hint-efi=hd0,msdos5 --hint-baremetal=ahci0,msdos5  4361d0d6-d\
a19-4e0a-ab8c-6e1bbfaf7e2c
                else
                  search --no-floppy --fs-uuid --set=root 4361d0d6-da19-\
4e0a-ab8c-6e1bbfaf7e2c
                fi
                echo        'Loading Linux 5.10.60-prod01 ...'
                linux        /boot/vmlinuz-5.10.60-prod01 root=UUID=4361\
d0d6-da19-4e0a-ab8c-6e1bbfaf7e2c ro  quiet splash 3 $vt_handoff console=\
ttyS0 console=tty0 ignore_loglevel_

    Minimum Emacs-like screen editing is supported. TAB lists
    completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a
    command-line or ESC to discard edits and return to the GRUB
    menu.
```

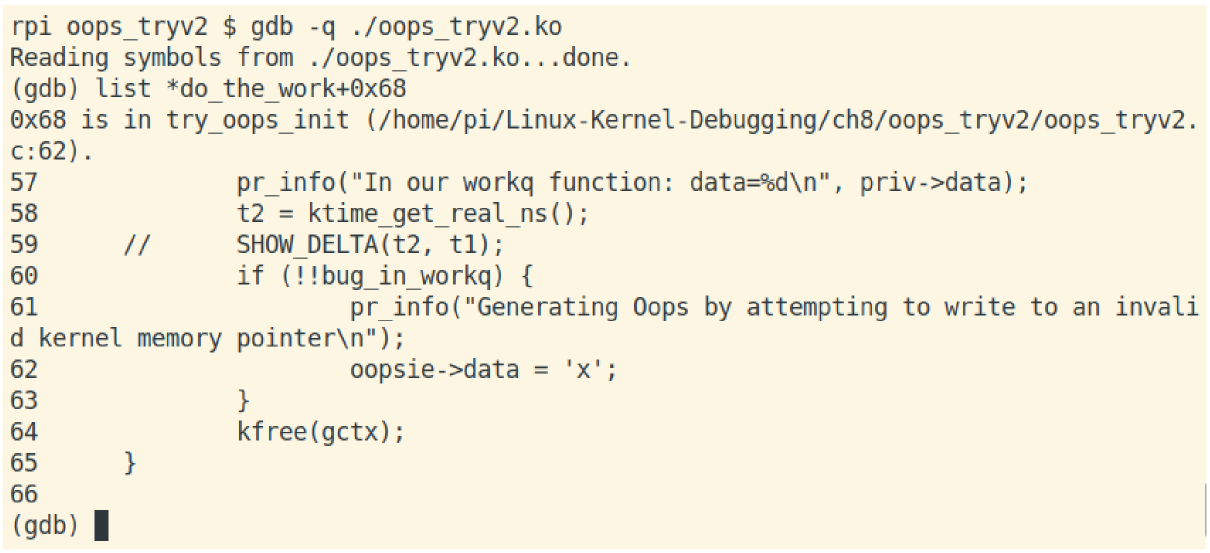


```
[  770.407919] BUG: kernel NULL pointer dereference, address: 0000000000000100
[  770.408580] #PF: supervisor write access in kernel mode
[  770.409050] #PF: error_code(0x0002) - not-present page
[  770.409521] PGD 0 P4D 0
[  770.409757] Oops: 0002 [#1] PREEMPT SMP PTI
[  770.410143] CPU: 1 PID: 1699 Comm: insmod Tainted: G           OE     5.10.60-prod01 #6
[  770.410869] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[  770.411615] RIP: 0010:irq_work+0x36/0x150 [oops_inirqv3]
[  770.412100] Code: 05 6f fb 9a 3f a9 00 01 ff 00 74 19 a9 00 00 0f 00 0f 84 f5 00 00 00 ba 48 00 00 00 f6
 c4 ff 0f 84 e7 00 00 00 0f 1f 44 00 00 <c7> 04 25 00 01 00 00 78 00 00 00 c3 55 65 4c 8b 04 25 c0 7b 01 00
[  770.413793] RSP: 0018:ffff9cc4800f0f78 EFLAGS: 00010006
[  770.414274] RAX: 0000000080010001 RBX: ffffffffc066b480 RCX: 000000000000080b
[  770.414922] RDX: 0000000000000068 RSI: ffffffffb4f27968 RDI: ffffffffc066b480
[  770.415546] RBP: ffff9cc4800f0f98 R08: 0000000000000000 R09: 0000000000000000
[  770.416168] R10: 0000000000000000 R11: 0000000000000000 R12: 0000000000000022
[  770.416787] R13: 0000000000000020 R14: 0000000000000000 R15: 0000000000000000
[  770.417397] FS:  00007f514a975540(0000) GS:ffff903c7dc40000(0000) knlGS:0000000000000000
[  770.418136] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[  770.418662] CR2: 0000000000000100 CR3: 00000000265fe006 CR4: 00000000000706e0
[  770.419283] Call Trace:
[  770.419500]  <IRQ>
[  770.419684]  ? irq_work_single+0x34/0x50
[  770.420032]  irq_work_run_list+0x31/0x50
[  770.420398]  irq_work_run+0x5a/0xf0
[  770.420711]  __sysvec_irq_work+0x30/0xd0
[  770.421085]  asm_call_irq_on_stack+0x12/0x20
[  770.421479]  </IRQ>
[  770.421677]  sysvec_irq_work+0x9f/0xc0
[  770.422478]  asm_sysvec_irq_work+0x12/0x20
[  770.423215] RIP: 0010:native_write_msr+0x6/0x30
[  770.423990] Code: 0f 1f 40 00 0f 1f 44 00 00 55 48 89 e5 0f 0b 48 c7 c7 60 07 07 b5 e8 51 70 c1 00 66 0f
 1f 84 00 00 00 00 00 89 f9 89 f0 0f 30 <0f> 1f 44 00 00 c3 55 48 c1 e2 20 89 f6 48 09 d6 31 d2 48 89 e5 e8
[  770.426876] RSP: 0018:ffff9cc482603bb0 EFLAGS: 00000206
[  770.427732] RAX: 00000000000000f6 RBX: 0000000000000010 RCX: 000000000000083f
[  770.429039] RDX: 0000000000000000 RSI: 00000000000000f6 RDI: 000000000000083f
[  770.430134] RBP: ffff9cc482603bb8 R08: 0000000000000010 R09: ffff903c137550a0
[  770.431195] R10: ffff903c01ae5410 R11: 0000000000000000 R12: ffffffffc066b480
[  770.432229] R13: 00000000000288a8 R14: 0000000000000001 R15: ffffffffc066b0d8
[  770.433264]  ? native_apic_msr_write+0x2b/0x30
[  770.434066]  x2apic_send_IPI_self+0x20/0x30
[  770.434859]  arch_irq_work_raise+0x2a/0x40
[  770.435620]  __irq_work_queue_local+0xbf/0x130
[  770.436398]  irq_work_queue+0x32/0x50
[  770.437091]  ? 0xffffffffc0662000
[  770.437751]  try_oops_init+0x2a/0x1000 [oops_inirqv3]
[  770.438559]  do_one_initcall+0x48/0x210
[  770.439353]  ? kmem_cache_alloc_trace+0x3ae/0x450
[  770.440431]  do_init_module+0x62/0x240
[  770.441121]  load_module+0x2a04/0x3080
[  770.441814]  ? security_kernel_post_read_file+0x5c/0x70
[  770.442651]  __do_sys_finit_module+0xc2/0x120
[  770.443390]  ? __do_sys_finit_module+0xc2/0x120
[  770.444141]  __x64_sys_finit_module+0x1a/0x20
```

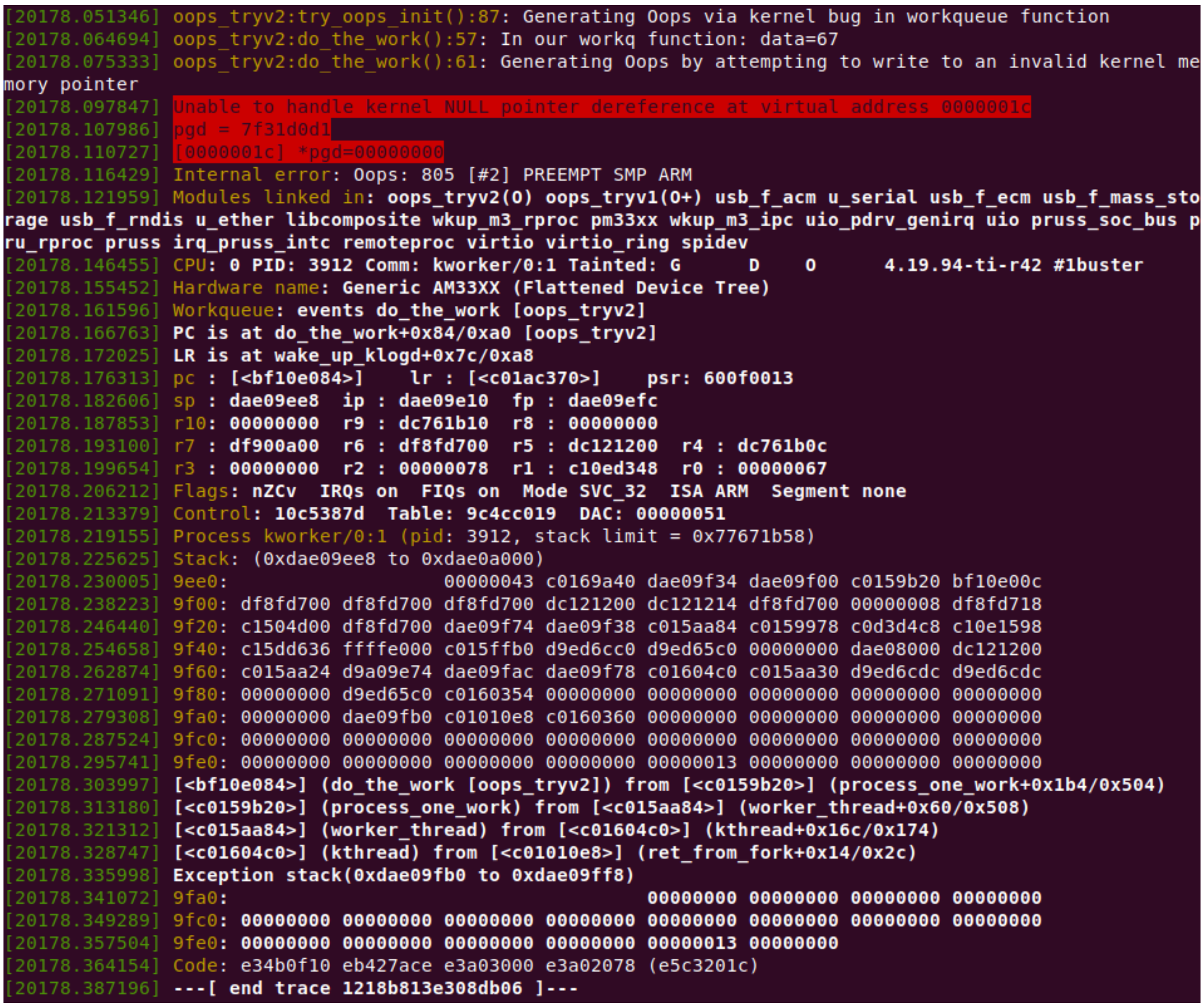


```
rpi oops_tryv2 #
rpi oops_tryv2 # modprobe netconsole netconsole=@192.168.1.24/wlan0,@192.168.1.101/
rpi oops_tryv2 # echo test123 > /dev/kmsg
rpi oops_tryv2 #
rpi oops_tryv2 # insmod ./oops_tryv2.ko bug_in_workq=yes
```

```
[ 6964.642063] test123
[ 6982.109243] oops_tryv2: loading out-of-tree module taints kernel.
[ 6982.115208] oops_tryv2:try_oops_init():87: Generating Oops via kernel bug in workqueue f
unction
[ 6982.127430] oops_tryv2:do_the_work():57: In our workq function: data=67
[ 6982.131918] oops_tryv2:do_the_work():61: Generating Oops by attempting to write to an in
valid kernel memory pointer
[ 6982.140055] 8<--- cut here ---
[ 6982.144180] Unable to handle kernel NULL pointer dereference at virtual address 0000001c
[ 6982.152208] pgd = 01cf7cd3
[ 6982.156062] [0000001c] *pgd=00000000
[ 6982.159842] Internal error: Oops: 817 [#1] ARM
[ 6982.163651] Modules linked in: oops_tryv2(O) netconsole aes_arm aes_generic cmac bnep hc
i_uart btbcm bluetooth ecdh_generic ecc libaes 8021q garp stp llc brcmfmac brcmutil sha256_
generic libsha256 cfg80211 rfkill raspberrypi_hwmon bcm2835_codec(C) bcm2835_isp(C) snd_bcm
2835(C) bcm2835_v4l2(C) v4l2_mem2mem bcm2835_mmal_vchiq(C) videobuf2_vmalloc videobuf2_dma_
contig videobuf2_memops videobuf2_v4l2 snd_pcm videobuf2_common vc_sm_cma(C) snd_timer snd
videodev mc uio_pdrv_genirq uio fixed i2c_dev ip_tables x_tables ipv6 [last unloaded: netco
nsole]
[ 6982.189999] CPU: 0 PID: 994 Comm: kworker/0:1 Tainted: G        WC O      5.10.17+ #1414
[ 6982.197569] Hardware name: BCM2835
[ 6982.201486] Workqueue: events do_the_work [oops_tryv2]
[ 6982.205388] PC is at do_the_work+0x68/0x94 [oops_tryv2]
[ 6982.209269] LR is at 0x0
[ 6982.213225] pc : [<bf1a0068>]    lr : [<00000000>]    psr: 60000013
```

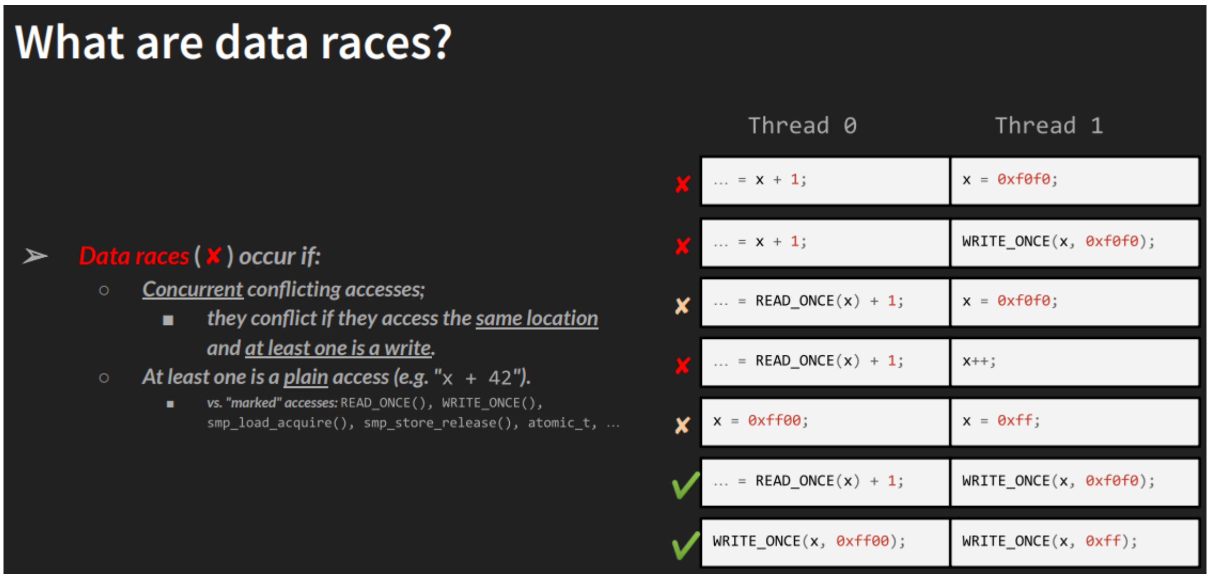


```
rpi oops_tryv2 $ gdb -q ./oops_tryv2.ko
Reading symbols from ./oops_tryv2.ko...done.
(gdb) list *do_the_work+0x68
0x68 is in try_oops_init (/home/pi/Linux-Kernel-Debugging/ch8/oops_tryv2/oops_tryv2.
c:62).
57              pr_info("In our workq function: data=%d\n", priv->data);
58              t2 = ktime_get_real_ns();
59      //      SHOW_DELTA(t2, t1);
60              if (!!bug_in_workq) {
61                      pr_info("Generating Oops by attempting to write to an invali
d kernel memory pointer\n");
62                      oopsie->data = 'x';
63              }
64              kfree(gctx);
65      }
66
(gdb) 
```



```
[20178.051346] oops_tryv2:try_oops_init():87: Generating Oops via kernel bug in workqueue function
[20178.064694] oops_tryv2:do_the_work():57: In our workq function: data=67
[20178.075333] oops_tryv2:do_the_work():61: Generating Oops by attempting to write to an invalid kernel me
mory pointer
[20178.097847] Unable to handle kernel NULL pointer dereference at virtual address 0000001c
[20178.107986] pgd = 7f31d0d1
[20178.110727] [0000001c] *pgd=00000000
[20178.116429] Internal error: Oops: 805 [#2] PREEMPT SMP ARM
[20178.121959] Modules linked in: oops_tryv2(O) oops_tryv1(O+) usb_f_acm u_serial usb_f_ecm usb_f_mass_sto
rage usb_f_rndis u_ether libcomposite wkup_m3_rproc pm33xx wkup_m3_ipc uio_pdrv_genirq uio pruss_soc_bus p
ru_rproc pruss irq_pruss_intc remoteproc virtio virtio_ring spidev
[20178.146455] CPU: 0 PID: 3912 Comm: kworker/0:1 Tainted: G      D    O      4.19.94-ti-r42 #1buster
[20178.155452] Hardware name: Generic AM33XX (Flattened Device Tree)
[20178.161596] Workqueue: events do_the_work [oops_tryv2]
[20178.166763] PC is at do_the_work+0x84/0xa0 [oops_tryv2]
[20178.172025] LR is at wake_up_klogd+0x7c/0xa8
[20178.176313] pc : [<bf10e084>]    lr : [<c01ac370>]    psr: 600f0013
[20178.182606] sp : dae09ee8  ip : dae09e10  fp : dae09efc
[20178.187853] r10: 00000000  r9 : dc761b10  r8 : 00000000
[20178.193100] r7 : df900a00  r6 : df8fd700  r5 : dc121200  r4 : dc761b0c
[20178.199654] r3 : 00000000  r2 : 00000078  r1 : c10ed348  r0 : 00000067
[20178.206212] Flags: nZCv  IRQs on  FIQs on  Mode SVC_32  ISA ARM  Segment none
[20178.213379] Control: 10c5387d  Table: 9c4cc019  DAC: 00000051
[20178.219155] Process kworker/0:1 (pid: 3912, stack limit = 0x77671b58)
[20178.225625] Stack: (0xdae09ee8 to 0xdae0a000)
[20178.230005] 9ee0:                   00000043 c0169a40 dae09f34 dae09f00 c0159b20 bf10e00c
[20178.238223] 9f00: df8fd700 df8fd700 df8fd700 dc121200 dc121214 df8fd700 00000008 df8fd718
[20178.246440] 9f20: c1504d00 df8fd700 dae09f74 dae09f38 c015aa84 c0159978 c0d3d4c8 c10e1598
[20178.254658] 9f40: c15dd636 ffffe000 c015ffb0 d9ed6cc0 d9ed65c0 00000000 dae08000 dc121200
[20178.262874] 9f60: c015aa24 d9a09e74 dae09fac dae09f78 c01604c0 c015aa30 d9ed6cdc d9ed6cdc
[20178.271091] 9f80: 00000000 d9ed65c0 c0160354 00000000 00000000 00000000 00000000 00000000
[20178.279308] 9fa0: 00000000 dae09fb0 c01010e8 c0160360 00000000 00000000 00000000 00000000
[20178.287524] 9fc0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
[20178.295741] 9fe0: 00000000 00000000 00000000 00000000 00000013 00000000 00000000 00000000
[20178.303997] [<bf10e084>] (do_the_work [oops_tryv2]) from [<c0159b20>] (process_one_work+0x1b4/0x504)
[20178.313180] [<c0159b20>] (process_one_work) from [<c015aa84>] (worker_thread+0x60/0x508)
[20178.321312] [<c015aa84>] (worker_thread) from [<c01604c0>] (kthread+0x16c/0x174)
[20178.328747] [<c01604c0>] (kthread) from [<c01010e8>] (ret_from_fork+0x14/0x2c)
[20178.335998] Exception stack(0xdae09fb0 to 0xdae09ff8)
[20178.341072] 9fa0:                                     00000000 00000000 00000000 00000000
[20178.349289] 9fc0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
[20178.357504] 9fe0: 00000000 00000000 00000000 00000000 00000013 00000000
[20178.364154] Code: e34b0f10 eb427ace e3a03000 e3a02078 (e5c3201c)
[20178.387196] ---[ end trace 1218b813e308db06 ]---
```

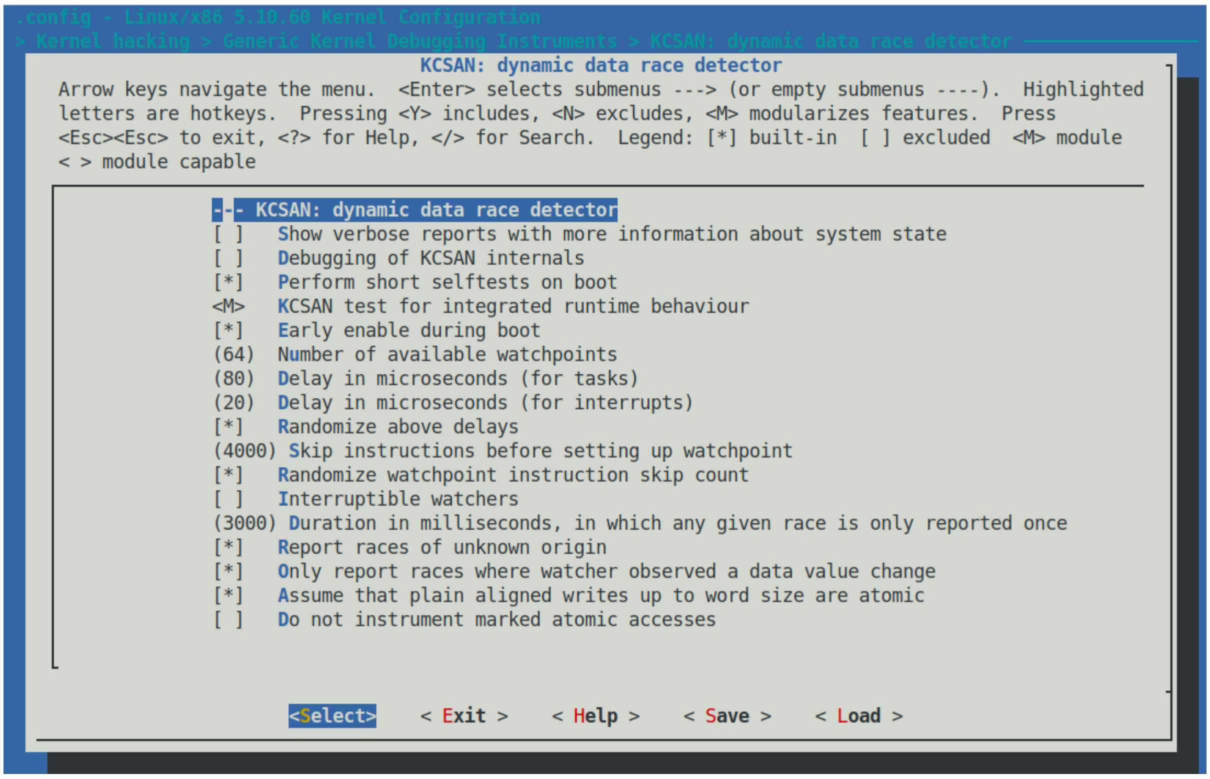
Chapter 8: Lock Debugging



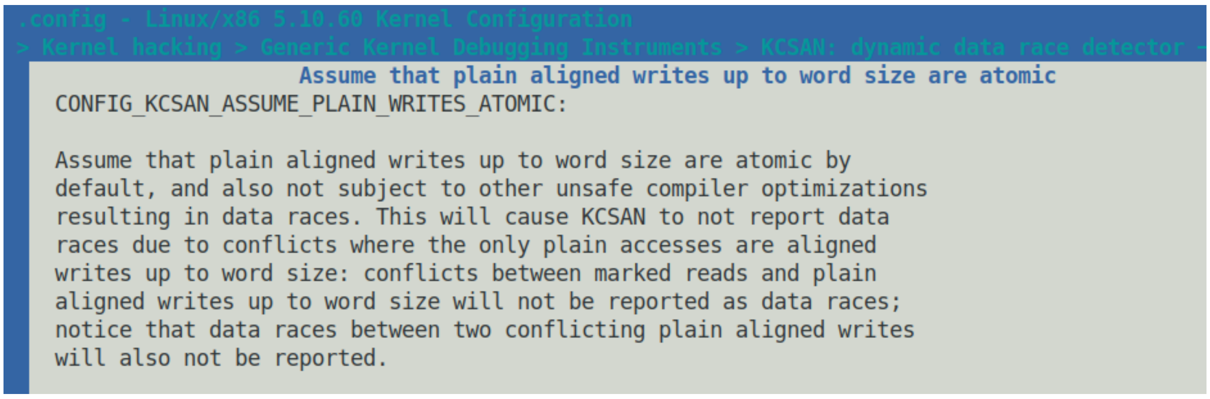
What are data races?

- ➢ *Data races* (✗) *occur if:*
  - ○ *Concurrent conflicting accesses;*
    - ■ *they conflict if they access the same location and at least one is a write.*
  - ○ *At least one is a plain access (e.g. "`x + 42`").*
    - ■ *vs. "marked" accesses:* `READ_ONCE(), WRITE_ONCE(), smp_load_acquire(), smp_store_release(), atomic_t, …`

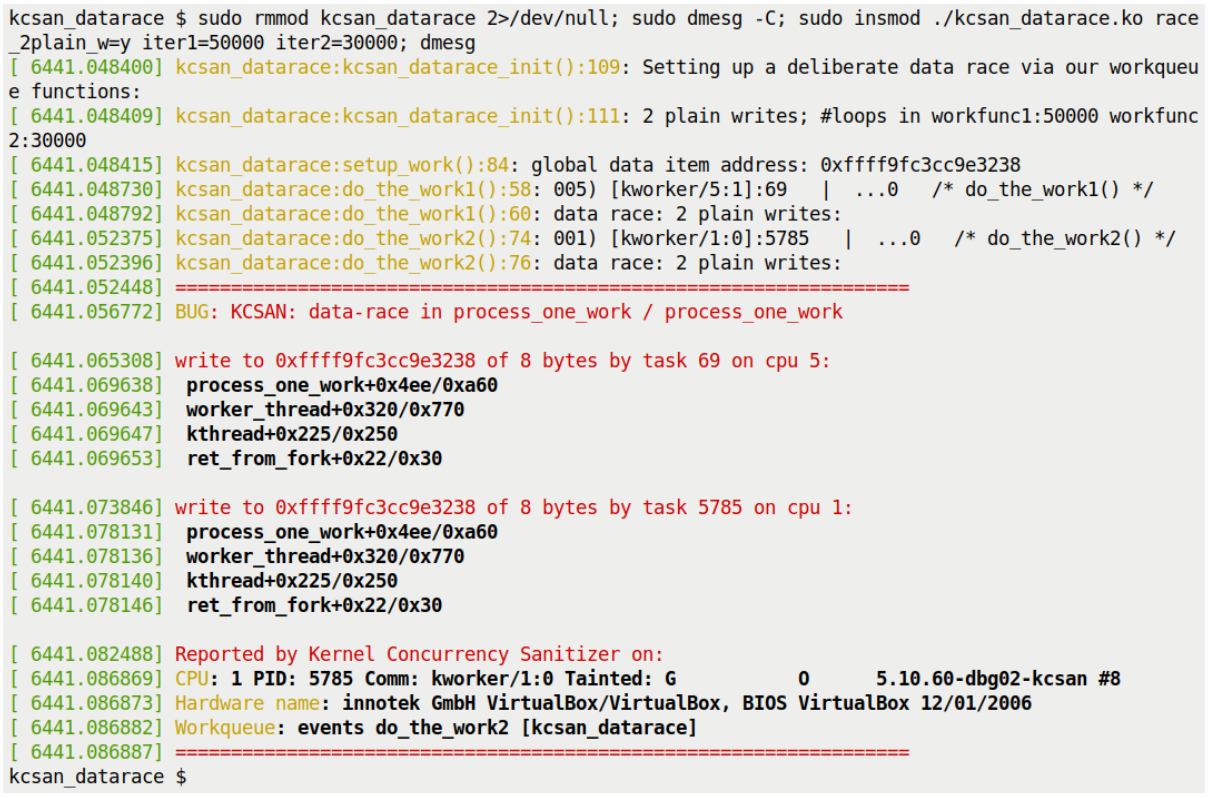
	Thread 0	Thread 1
✗	`… = x + 1;`	`x = 0xf0f0;`
✗	`… = x + 1;`	`WRITE_ONCE(x, 0xf0f0);`
✗	`… = READ_ONCE(x) + 1;`	`x = 0xf0f0;`
✗	`… = READ_ONCE(x) + 1;`	`x++;`
✗	`x = 0xff00;`	`x = 0xff;`
✓	`… = READ_ONCE(x) + 1;`	`WRITE_ONCE(x, 0xf0f0);`
✓	`WRITE_ONCE(x, 0xff00);`	`WRITE_ONCE(x, 0xff);`



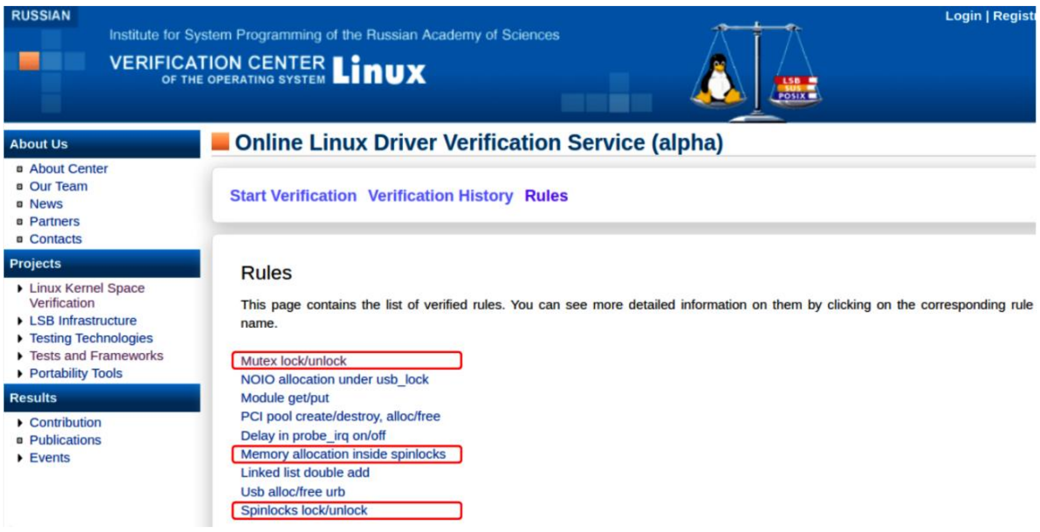
```
.config - Linux/x86 5.10.60 Kernel Configuration
> Kernel hacking > Generic Kernel Debugging Instruments > KCSAN: dynamic data race detector ──────
                          KCSAN: dynamic data race detector
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted
  letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press
  <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module
  < > module capable

              --- KCSAN: dynamic data race detector
              [ ]   Show verbose reports with more information about system state
              [ ]   Debugging of KCSAN internals
              [*]   Perform short selftests on boot
              <M>   KCSAN test for integrated runtime behaviour
              [*]   Early enable during boot
              (64)  Number of available watchpoints
              (80)  Delay in microseconds (for tasks)
              (20)  Delay in microseconds (for interrupts)
              [*]   Randomize above delays
              (4000) Skip instructions before setting up watchpoint
              [*]   Randomize watchpoint instruction skip count
              [ ]   Interruptible watchers
              (3000) Duration in milliseconds, in which any given race is only reported once
              [*]   Report races of unknown origin
              [*]   Only report races where watcher observed a data value change
              [*]   Assume that plain aligned writes up to word size are atomic
              [ ]   Do not instrument marked atomic accesses

          <Select>    < Exit >    < Help >    < Save >    < Load >
```



```
.config - Linux/x86 5.10.60 Kernel Configuration
> Kernel hacking > Generic Kernel Debugging Instruments > KCSAN: dynamic data race detector ─
                      Assume that plain aligned writes up to word size are atomic
  CONFIG_KCSAN_ASSUME_PLAIN_WRITES_ATOMIC:

  Assume that plain aligned writes up to word size are atomic by
  default, and also not subject to other unsafe compiler optimizations
  resulting in data races. This will cause KCSAN to not report data
  races due to conflicts where the only plain accesses are aligned
  writes up to word size: conflicts between marked reads and plain
  aligned writes up to word size will not be reported as data races;
  notice that data races between two conflicting plain aligned writes
  will also not be reported.
```



```
kcsan_datarace $ sudo rmmod kcsan_datarace 2>/dev/null; sudo dmesg -C; sudo insmod ./kcsan_datarace.ko race
_2plain_w=y iter1=50000 iter2=30000; dmesg
[ 6441.048400] kcsan_datarace:kcsan_datarace_init():109: Setting up a deliberate data race via our workqueu
e functions:
[ 6441.048409] kcsan_datarace:kcsan_datarace_init():111: 2 plain writes; #loops in workfunc1:50000 workfunc
2:30000
[ 6441.048415] kcsan_datarace:setup_work():84: global data item address: 0xffff9fc3cc9e3238
[ 6441.048730] kcsan_datarace:do_the_work1():58: 005) [kworker/5:1]:69   |  ...0   /* do_the_work1() */
[ 6441.048792] kcsan_datarace:do_the_work1():60: data race: 2 plain writes:
[ 6441.052375] kcsan_datarace:do_the_work2():74: 001) [kworker/1:0]:5785   |  ...0   /* do_the_work2() */
[ 6441.052396] kcsan_datarace:do_the_work2():76: data race: 2 plain writes:
[ 6441.052448] ==================================================================
[ 6441.056772] BUG: KCSAN: data-race in process_one_work / process_one_work

[ 6441.065308] write to 0xffff9fc3cc9e3238 of 8 bytes by task 69 on cpu 5:
[ 6441.069638]  process_one_work+0x4ee/0xa60
[ 6441.069643]  worker_thread+0x320/0x770
[ 6441.069647]  kthread+0x225/0x250
[ 6441.069653]  ret_from_fork+0x22/0x30

[ 6441.073846] write to 0xffff9fc3cc9e3238 of 8 bytes by task 5785 on cpu 1:
[ 6441.078131]  process_one_work+0x4ee/0xa60
[ 6441.078136]  worker_thread+0x320/0x770
[ 6441.078140]  kthread+0x225/0x250
[ 6441.078146]  ret_from_fork+0x22/0x30

[ 6441.082488] Reported by Kernel Concurrency Sanitizer on:
[ 6441.086869] CPU: 1 PID: 5785 Comm: kworker/1:0 Tainted: G           O      5.10.60-dbg02-kcsan #8
[ 6441.086873] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[ 6441.086882] Workqueue: events do_the_work2 [kcsan_datarace]
[ 6441.086887] ==================================================================
kcsan_datarace $
```



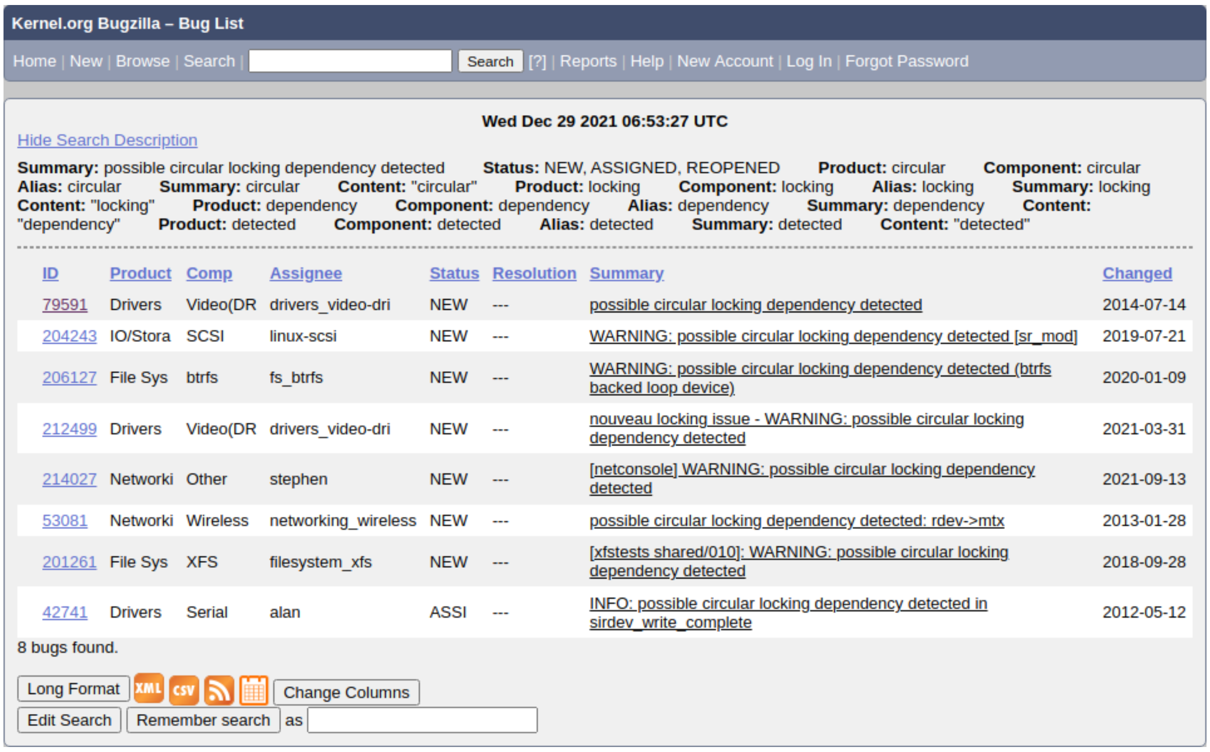
Online Linux Driver Verification Service (alpha)

Start Verification Verification History Rules

Rules

This page contains the list of verified rules. You can see more detailed information on them by clicking on the corresponding rule name.

Mutex lock/unlock
NOIO allocation under usb_lock
Module get/put
PCI pool create/destroy, alloc/free
Delay in probe_irq on/off
Memory allocation inside spinlocks
Linked list double add
Usb alloc/free urb
Spinlocks lock/unlock


Kernel.org Bugzilla – Bug List
Home | New | Browse | Search | Search [?] | Reports | Help | New Account | Log In | Forgot Password

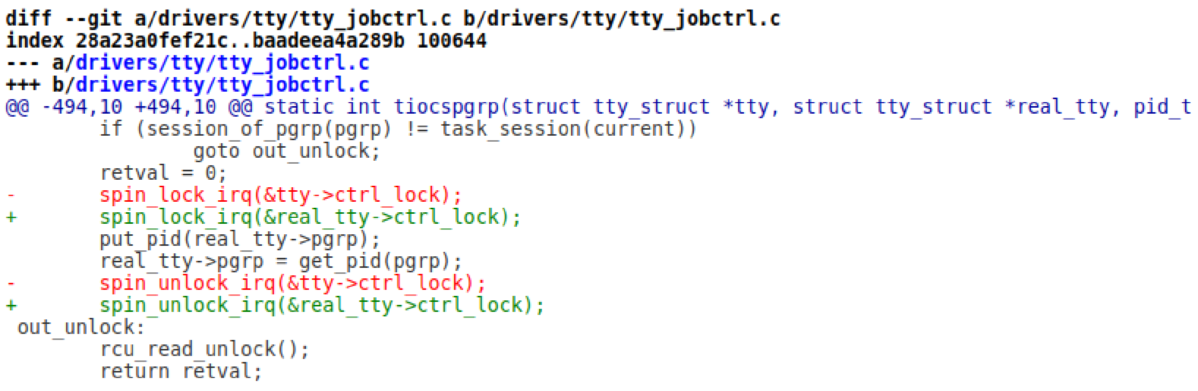
Wed Dec 29 2021 06:53:27 UTC

Hide Search Description

Summary: possible circular locking dependency detected **Status:** NEW, ASSIGNED, REOPENED **Product:** circular **Component:** circular **Alias:** circular **Summary:** circular **Content:** "circular" **Product:** locking **Component:** locking **Alias:** locking **Summary:** locking **Content:** "locking" **Product:** dependency **Component:** dependency **Alias:** dependency **Summary:** dependency **Content:** "dependency" **Product:** detected **Component:** detected **Alias:** detected **Summary:** detected **Content:** "detected"

ID	Product	Comp	Assignee	Status	Resolution	Summary	Changed
79591	Drivers	Video(DR	drivers_video-dri	NEW	---	possible circular locking dependency detected	2014-07-14
204243	IO/Stora	SCSI	linux-scsi	NEW	---	WARNING: possible circular locking dependency detected [sr_mod]	2019-07-21
206127	File Sys	btrfs	fs_btrfs	NEW	---	WARNING: possible circular locking dependency detected (btrfs backed loop device)	2020-01-09
212499	Drivers	Video(DR	drivers_video-dri	NEW	---	nouveau locking issue - WARNING: possible circular locking dependency detected	2021-03-31
214027	Networki	Other	stephen	NEW	---	[netconsole] WARNING: possible circular locking dependency detected	2021-09-13
53081	Networki	Wireless	networking_wireless	NEW	---	possible circular locking dependency detected: rdev->mtx	2013-01-28
201261	File Sys	XFS	filesystem_xfs	NEW	---	[xfstests shared/010]: WARNING: possible circular locking dependency detected	2018-09-28
42741	Drivers	Serial	alan	ASSI	---	INFO: possible circular locking dependency detected in sirdev_write_complete	2012-05-12

8 bugs found.


```
diff --git a/drivers/tty/tty_jobctrl.c b/drivers/tty/tty_jobctrl.c
index 28a23a0fef21c..baadeea4a289b 100644
--- a/drivers/tty/tty_jobctrl.c
+++ b/drivers/tty/tty_jobctrl.c
@@ -494,10 +494,10 @@ static int tiocspgrp(struct tty_struct *tty, struct tty_struct *real_tty, pid_t
        if (session_of_pgrp(pgrp) != task_session(current))
                goto out_unlock;
        retval = 0;
-       spin_lock_irq(&tty->ctrl_lock);
+       spin_lock_irq(&real_tty->ctrl_lock);
        put_pid(real_tty->pgrp);
        real_tty->pgrp = get_pid(pgrp);
-       spin_unlock_irq(&tty->ctrl_lock);
+       spin_unlock_irq(&real_tty->ctrl_lock);
 out_unlock:
        rcu_read_unlock();
        return retval;
```

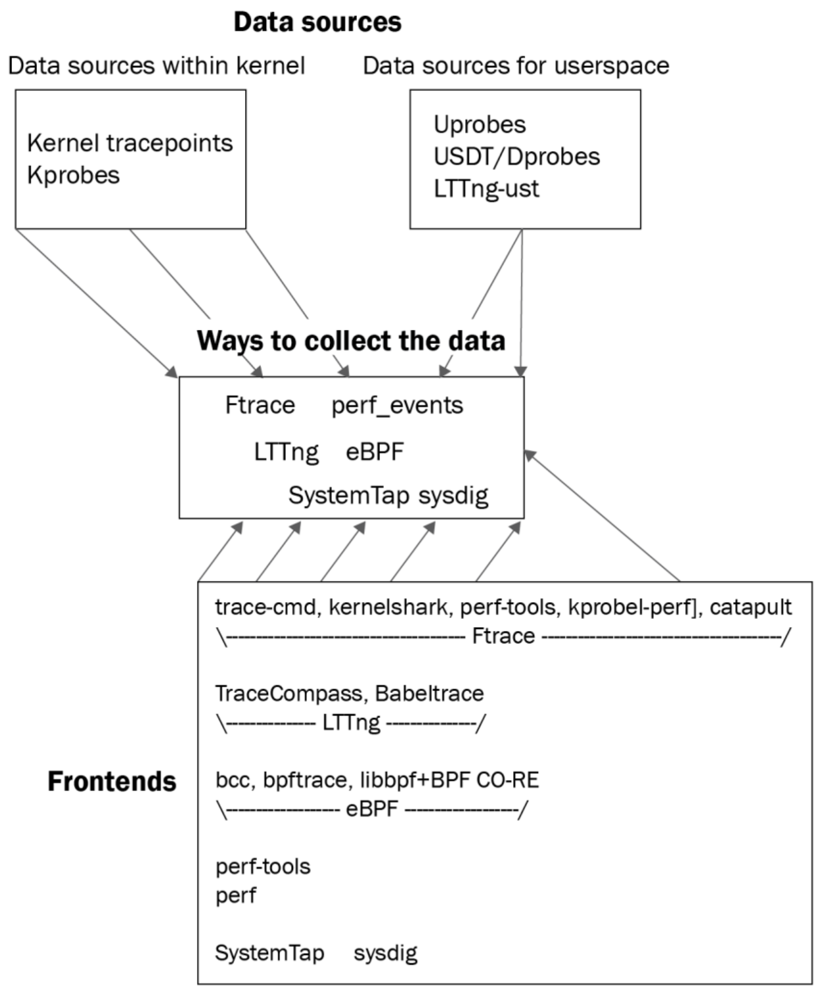
CONFIG_KCSAN_REPORT_ONCE_IN_MS	Rate limiting of data race reports (set to a duration of 3 s by default) to avoid flooding the console and log buffers with reports; set to 0 to disable rate limiting.	3000
CONFIG_KCSAN_REPORT_RACE_UNKNOWN_ORIGIN	If yes (default), report races where only one access is known (the others are unknown); only reported if the data value changed while delayed.	y
CONFIG_KCSAN_REPORT_VALUE_CHANGE_ONLY	Only report a data race when the data value changed (implying that, if a conflicting write was seen but the data value remained unchanged, don't report it).	y
CONFIG_KCSAN_ASSUME_PLAIN_WRITES_ATOMIC	If yes (the default), assume that plain aligned writes up to the processor word size are atomic; turning this off results in more reports (stricter mode). You'll find that this needs to be changed to n to test a simple two-plain-integer-writes data race.	y
CONFIG_KCSAN_IGNORE_ATOMICS	Don't check marked atomic accesses; has implications on what is reported as a data race (for example, this, along with CONFIG_KCSAN_REPORT_RACE_UNKNOWN_ORIGIN=n, implies that races where at least one access is marked atomic never get reported).	n

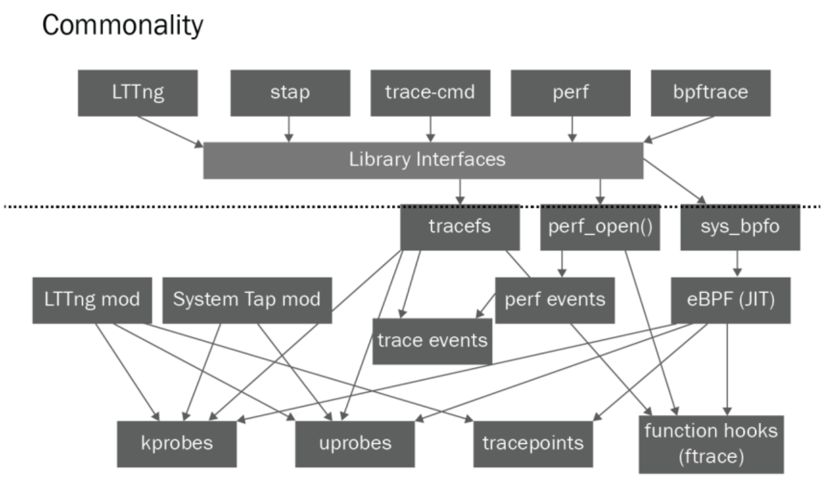
Kernel config (or kernel module parameter)	Meaning	Default value
`CONFIG_KCSAN_VERBOSE`	Show more in the report (including locks held and IRQ trace events); can cause instability.	`n`
`CONFIG_KCSAN_SELFTEST`	Runs KCSAN self-tests at boot time; kernel panics on failure.	`y`
`CONFIG_KCSAN_TEST`	Various KCSAN test cases (internally uses the kernel's KUnit and Torture test frameworks); can be built as a module by specifying `m` here.	`n`
`CONFIG_KCSAN_EARLY_ENABLE`	Enables KCSAN during early boot.	`y`
`CONFIG_KCSAN_UDELAY_TASK (or kcsan.udelay_task)`	Delay after setting up a watchpoint, for tasks (microseconds).	`80`
`CONFIG_KCSAN_UDELAY_INTERRUPT (or kcsan.udelay_interrupt)`	Delay after setting up a watchpoint, for interrupts (microseconds).	`20`
`CONFIG_KCSAN_SKIP_WATCH (or kcsan.skip_watch)`	Number of per-CPU memory operations to skip before it sets up another watchpoint; this tunable has the most impact on system performance and detecting data races. A smaller number implies better race detection with more degradation in system performance (and vice versa).	`4000`
`CONFIG_KCSAN_INTERRUPT_WATCHER (or kcsan.interrupt_watcher)`	When enabled, a task that set up a watchpoint can be interrupted while delayed allowing detection of races in this situation. Disabled by default (safer, else can generate false positives).	`n`

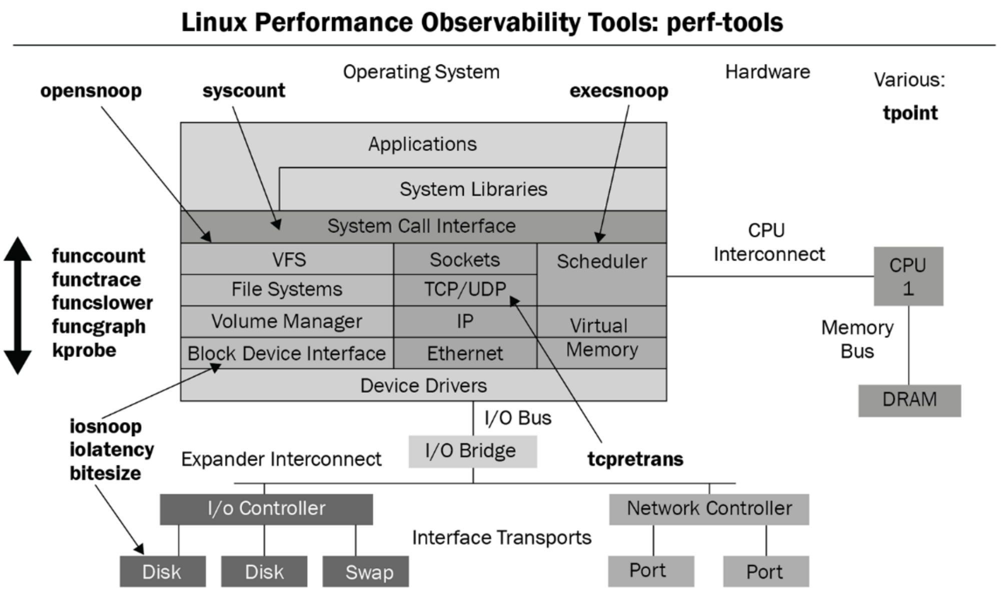
# loops in workfunc 1	# loops in workfunc 2	KCSAN catches the data race?
10,000	5,000	No
20,000	10,000	Yes
75,000	50,000	Yes

Action on `/sys/kernel/debug/kcsan`	Effect
Reading it	Shows statistics regarding KCSAN runtime; includes the number of watchpoints, data races detected, blacklisted functions, and so on
Writing `on` / `off` to it	Toggles KCSAN on/off
Writing `!funcname` to it	Blacklists reporting any data race where the function `funcname` is one of the top stack frames in either function involved in the race
Writing `blacklist`	Stop reporting frequently occurring data races
Writing `whitelist`	Keep reporting frequently occurring data races; helpful for testing/reproducing data races

Chapter 9: Tracing the Kernel Flow

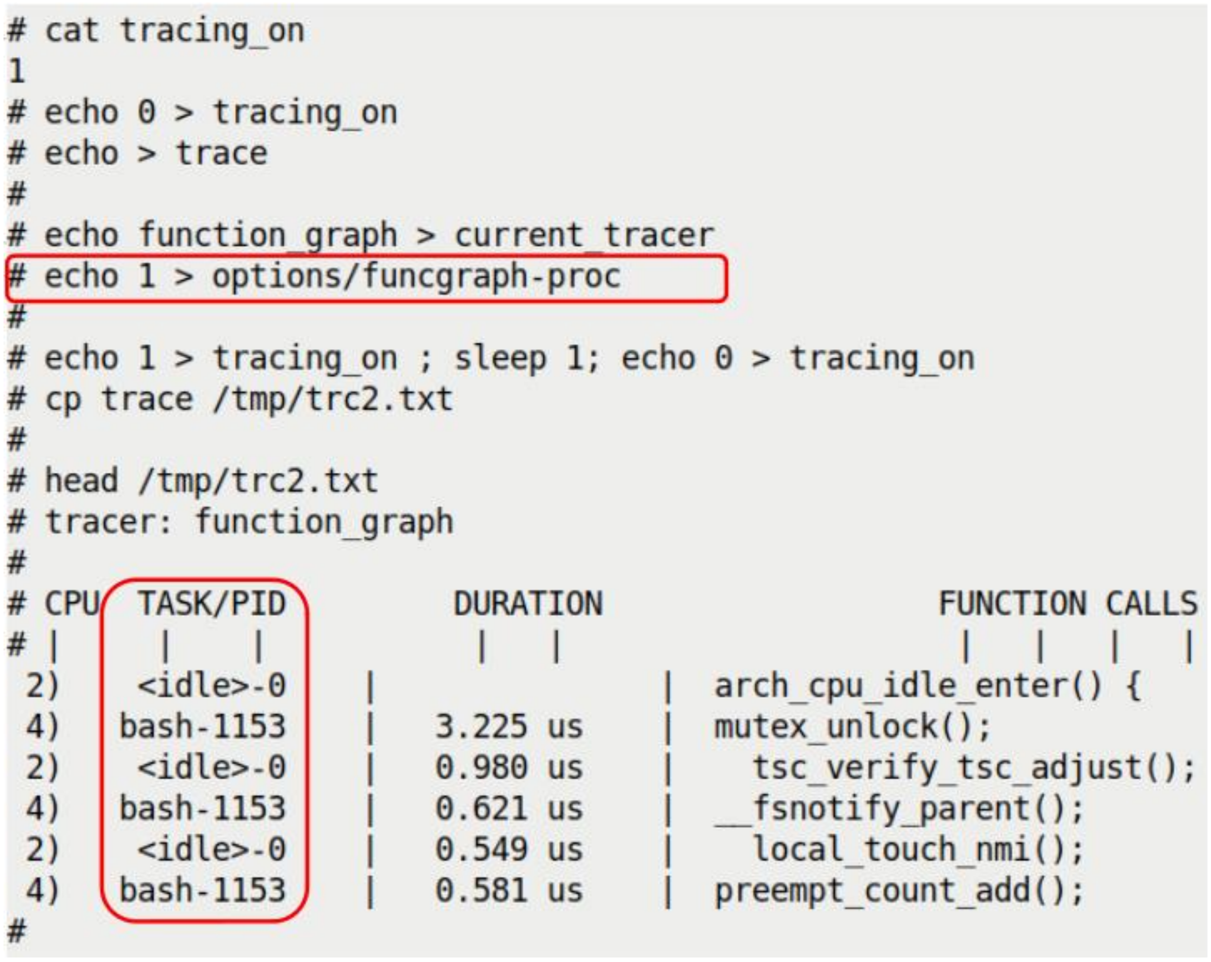

Data sources
Data sources within kernel
Data sources for userspace
Kernel tracepoints
Kprobes
Uprobes
USDT/Dprobes
LTTng-ust
Ways to collect the data
Ftrace
perf_events
LTTng
eBPF
SystemTap sysdig
Frontends
trace-cmd, kernelshark, perf-tools, kprobel-perf], catapult
------------------------------------ Ftrace ------------------------------------/
TraceCompass, Babeltrace
------------- LTTng -------------/
bcc, bpftrace, libbpf+BPF CO-RE
----------------- eBPF -----------------/
perf-tools
perf
SystemTap
sysdig



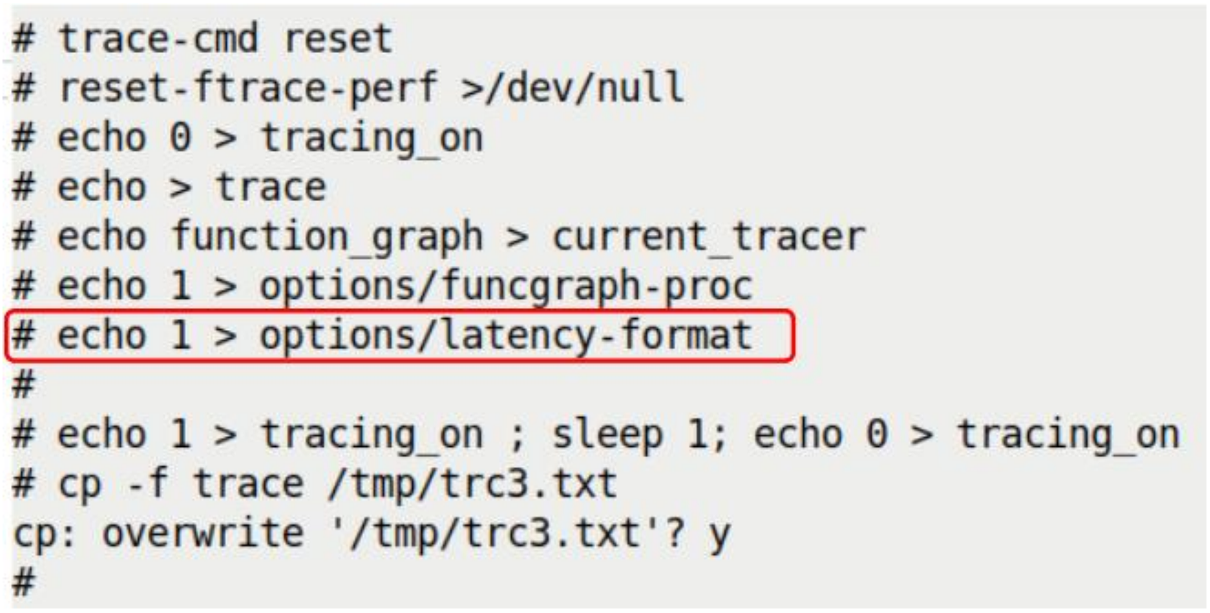


```
# pwd
/sys/kernel/tracing
# ls
available_events             max_graph_depth          stack_max_size
available_filter_functions   options/                 stack_trace
available_tracers            per_cpu/                 stack_trace_filter
buffer_percent               printk_formats           synthetic_events
buffer_size_kb               README                   timestamp_mode
buffer_total_size_kb         saved_cmdlines           trace
current_tracer               saved_cmdlines_size      trace_clock
dynamic_events               saved_tgids              trace_marker
dyn_ftrace_total_info        set_event                trace_marker_raw
enabled_functions            set_event_notrace_pid    trace_options
error_log                    set_event_pid            trace_pipe
events/                      set_ftrace_filter        trace_stat/
free_buffer                  set_ftrace_notrace       tracing_cpumask
function_profile_enabled     set_ftrace_notrace_pid   tracing_max_latency
hwlat_detector/              set_ftrace_pid           tracing_on
instances/                   set_graph_function       tracing_thresh
kprobe_events                set_graph_notrace        uprobe_events
kprobe_profile               snapshot                 uprobe_profile
#
```

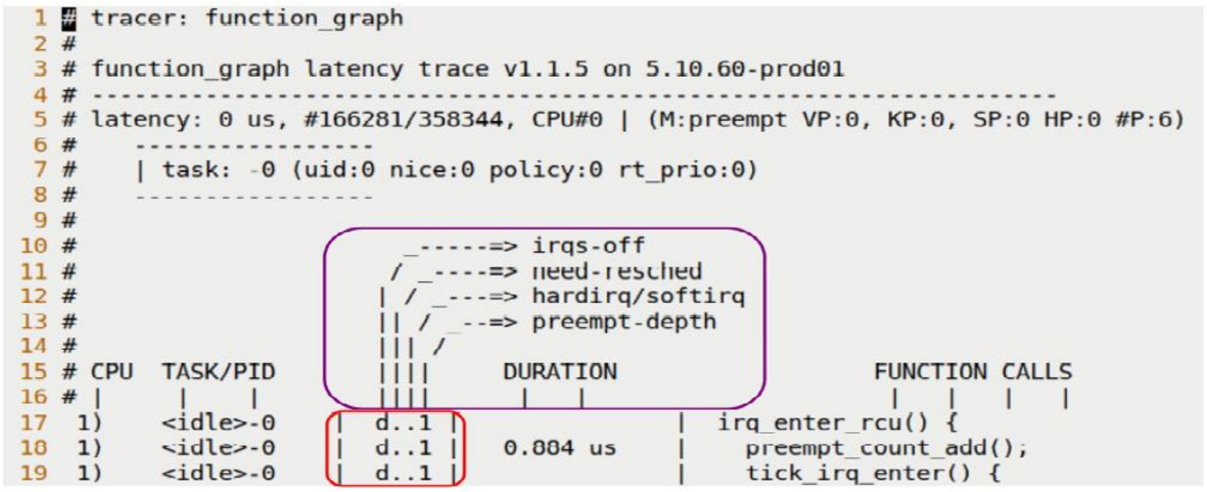
```
# pwd
/sys/kernel/tracing
# ls options/
annotate          event-fork          func_stack_trace  pause-on-trace   sym-offset
bin               funcgraph-abstime   function-fork     printk-msg-only  sym-userobj
blk_cgname        funcgraph-cpu       function-trace    print-parent     test_nop_accept
blk_cgroup        funcgraph-duration  graph-time        raw              test_nop_refuse
blk_classic       funcgraph-irqs      hex               record-cmd       trace_printk
block             funcgraph-overhead  irq-info          record-tgid      userstacktrace
context-info      funcgraph-overrun   latency-format    sleep-time       verbose
disable_on_free   funcgraph-proc      markers           stacktrace
display-graph     funcgraph-tail      overwrite         sym-addr
#
```



```
# cat tracing_on
1
# echo 0 > tracing_on
# echo > trace
#
# echo function_graph > current_tracer
# echo 1 > options/funcgraph-proc
#
# echo 1 > tracing_on ; sleep 1; echo 0 > tracing_on
# cp trace /tmp/trc2.txt
#
# head /tmp/trc2.txt
# tracer: function_graph
#
# CPU  TASK/PID         DURATION                  FUNCTION CALLS
# |     |    |           |   |                     |   |   |   |
 2)    <idle>-0    |               |  arch_cpu_idle_enter() {
 4)   bash-1153    |   3.225 us    |  mutex_unlock();
 2)    <idle>-0    |   0.980 us    |    tsc_verify_tsc_adjust();
 4)   bash-1153    |   0.621 us    |  __fsnotify_parent();
 2)    <idle>-0    |   0.549 us    |    local_touch_nmi();
 4)   bash-1153    |   0.581 us    |  preempt_count_add();
#
```

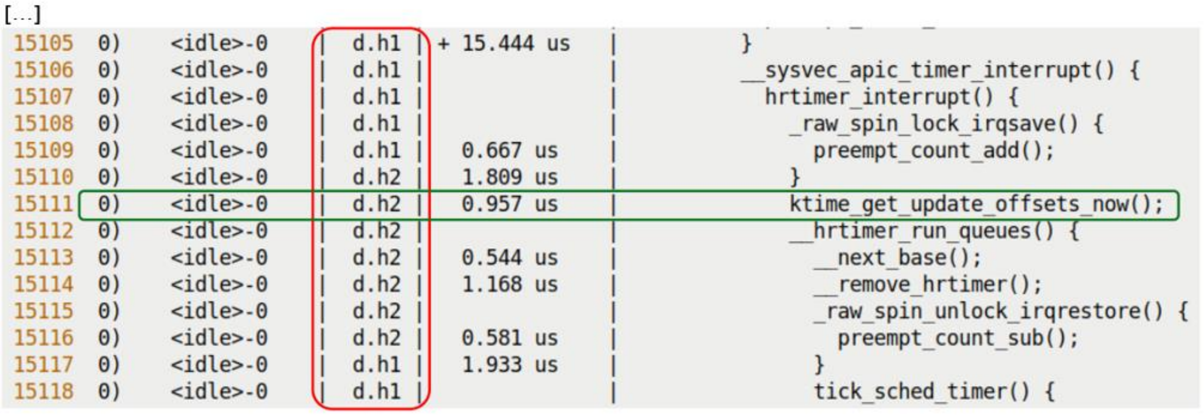


```
# trace-cmd reset
# reset-ftrace-perf >/dev/null
# echo 0 > tracing_on
# echo > trace
# echo function_graph > current_tracer
# echo 1 > options/funcgraph-proc
# echo 1 > options/latency-format
#
# echo 1 > tracing_on ; sleep 1; echo 0 > tracing_on
# cp -f trace /tmp/trc3.txt
cp: overwrite '/tmp/trc3.txt'? y
#
```



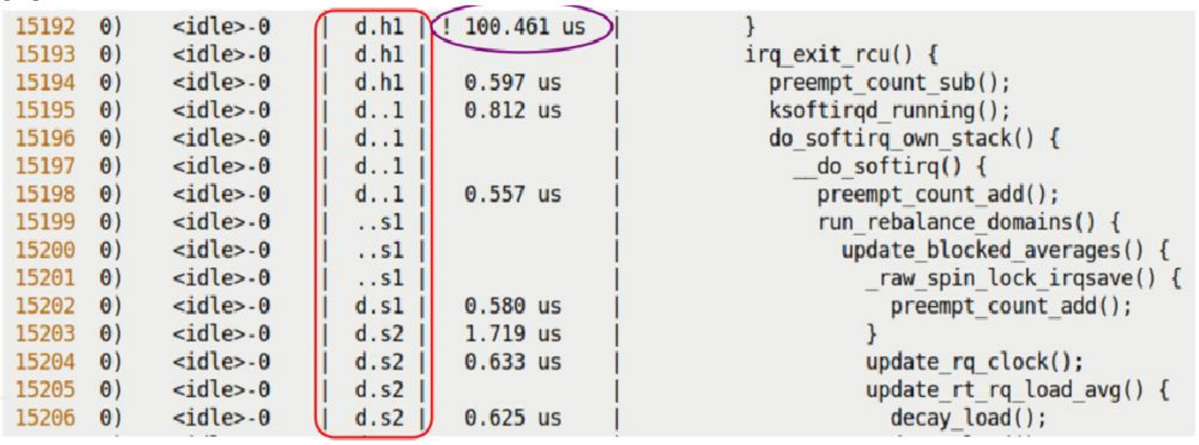
```
 1 # tracer: function_graph
 2 #
 3 # function_graph latency trace v1.1.5 on 5.10.60-prod01
 4 # --------------------------------------------------------------------
 5 # latency: 0 us, #166281/358344, CPU#0 | (M:preempt VP:0, KP:0, SP:0 HP:0 #P:6)
 6 #    -----------------
 7 #    | task: -0 (uid:0 nice:0 policy:0 rt_prio:0)
 8 #    -----------------
 9 #
10 #                         _-----=> irqs-off
11 #                        / _----=> need-resched
12 #                       | / _---=> hardirq/softirq
13 #                       || / _--=> preempt-depth
14 #                       ||| /
15 # CPU  TASK/PID         ||||     DURATION                   FUNCTION CALLS
16 # |     |    |          ||||      |   |                      |   |   |   |
17  1)    <idle>-0      |  d..1 |                    |  irq_enter_rcu() {
18  1)    <idle>-0      |  d..1 |    0.884 us        |    preempt_count_add();
19  1)    <idle>-0      |  d..1 |                    |    tick_irq_enter() {
```

[…]



```
15105  0)    <idle>-0     |  d.h1 | + 15.444 us   |          }
15106  0)    <idle>-0     |  d.h1 |               |          __sysvec_apic_timer_interrupt() {
15107  0)    <idle>-0     |  d.h1 |               |            hrtimer_interrupt() {
15108  0)    <idle>-0     |  d.h1 |               |              _raw_spin_lock_irqsave() {
15109  0)    <idle>-0     |  d.h1 |   0.667 us    |                preempt_count_add();
15110  0)    <idle>-0     |  d.h2 |   1.809 us    |              }
15111  0)    <idle>-0     |  d.h2 |   0.957 us    |              ktime_get_update_offsets_now();
15112  0)    <idle>-0     |  d.h2 |               |              __hrtimer_run_queues() {
15113  0)    <idle>-0     |  d.h2 |   0.544 us    |                __next_base();
15114  0)    <idle>-0     |  d.h2 |   1.168 us    |                __remove_hrtimer();
15115  0)    <idle>-0     |  d.h2 |               |                _raw_spin_unlock_irqrestore() {
15116  0)    <idle>-0     |  d.h2 |   0.581 us    |                  preempt_count_sub();
15117  0)    <idle>-0     |  d.h1 |   1.933 us    |                }
15118  0)    <idle>-0     |  d.h1 |               |                tick_sched_timer() {
```

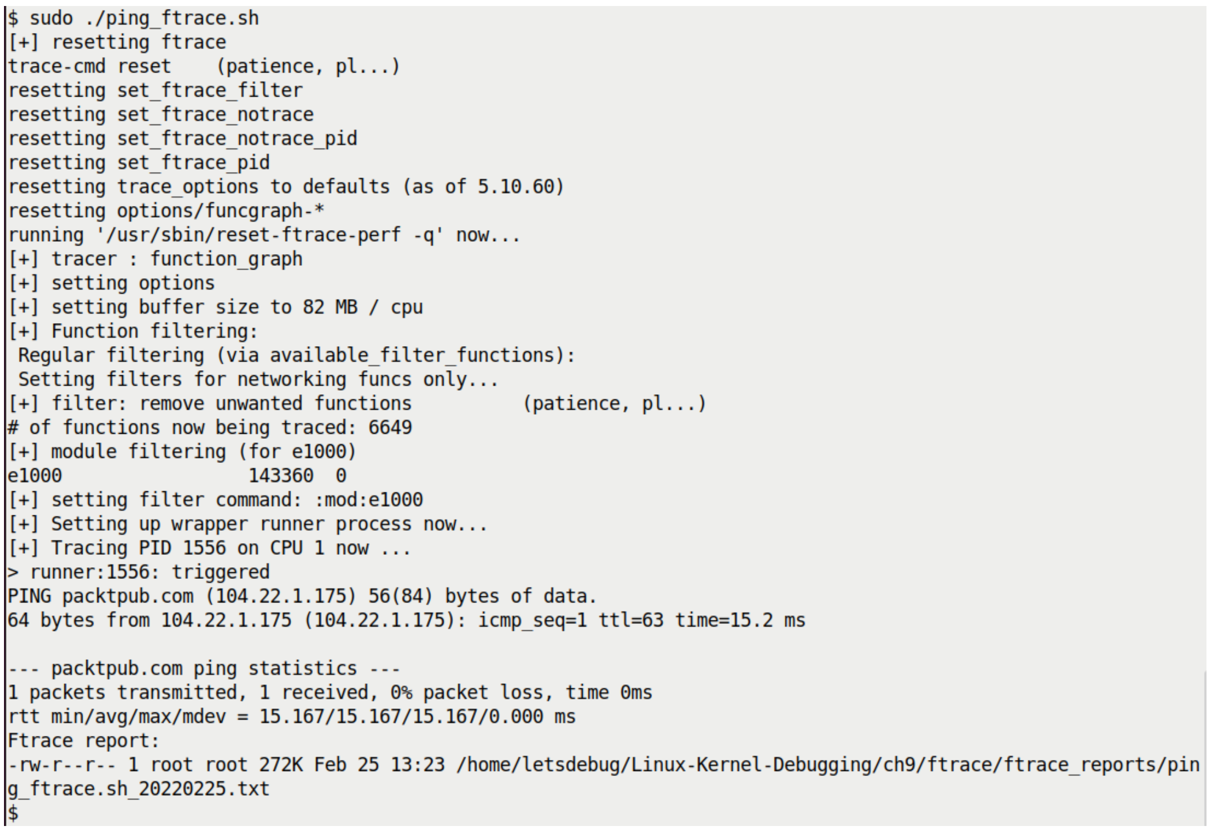
[…]



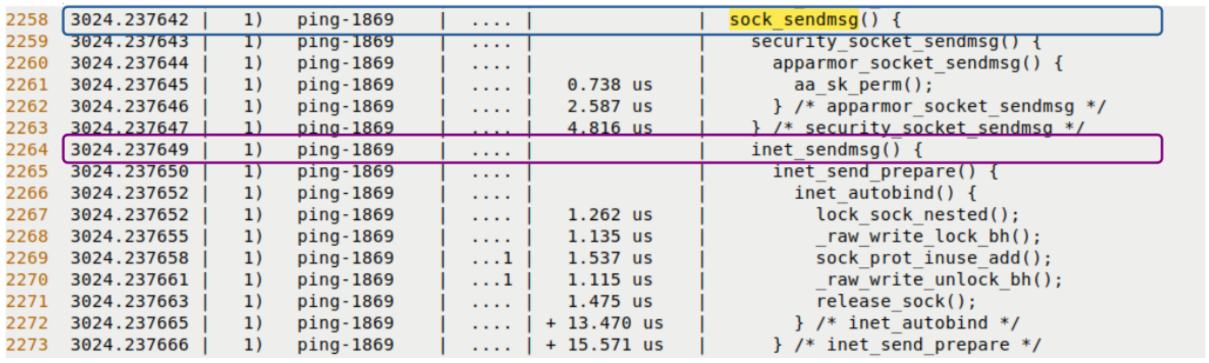
```
15192  0)    <idle>-0     |  d.h1 | ! 100.461 us  |          }
15193  0)    <idle>-0     |  d.h1 |               |          irq_exit_rcu() {
15194  0)    <idle>-0     |  d.h1 |   0.597 us    |            preempt_count_sub();
15195  0)    <idle>-0     |  d..1 |   0.812 us    |            ksoftirqd_running();
15196  0)    <idle>-0     |  d..1 |               |            do_softirq_own_stack() {
15197  0)    <idle>-0     |  d..1 |               |              __do_softirq() {
15198  0)    <idle>-0     |  d..1 |   0.557 us    |                preempt_count_add();
15199  0)    <idle>-0     |  ..s1 |               |                run_rebalance_domains() {
15200  0)    <idle>-0     |  ..s1 |               |                  update_blocked_averages() {
15201  0)    <idle>-0     |  ..s1 |               |                    _raw_spin_lock_irqsave() {
15202  0)    <idle>-0     |  d.s1 |   0.580 us    |                      preempt_count_add();
15203  0)    <idle>-0     |  d.s2 |   1.719 us    |                    }
15204  0)    <idle>-0     |  d.s2 |   0.633 us    |                    update_rq_clock();
15205  0)    <idle>-0     |  d.s2 |               |                    update_rt_rq_load_avg() {
15206  0)    <idle>-0     |  d.s2 |   0.625 us    |                      decay_load();
```

```
# cat trace_options
print-parent
nosym-offset
nosym-addr
noverbose
noraw
nohex
nobin
noblock
trace_printk
annotate
nouserstacktrace
nosym-userobj
noprintk-msg-only
context-info
nolatency-format
record-cmd
norecord-tgid
overwrite
nodisable_on_free
irq-info
markers
noevent-fork
nopause-on-trace
function-trace
nofunction-fork
nodisplay-graph
nostacktrace
notest_nop_accept
notest_nop_refuse
#
```

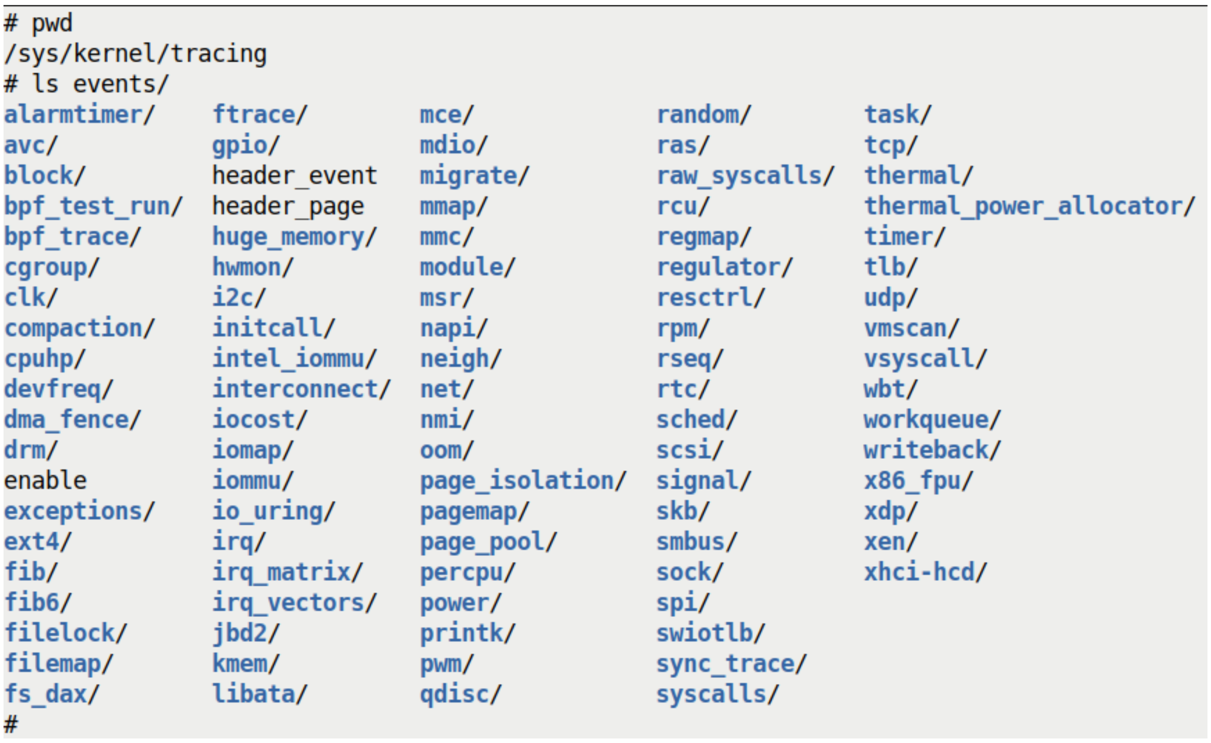
```
available_filter_functions - list of functions that can be filtered on
set_ftrace_filter     - echo function name in here to only trace these
                        functions
           accepts: func_full_name or glob-matching-pattern
           modules: Can select a group via module
            Format: :mod:<module-name>
           example: echo :mod:ext3 > set_ftrace_filter
          triggers: a command to perform when function is hit
            Format: <function>:<trigger>[:count]
           trigger: traceon, traceoff
                    enable_event:<system>:<event>
                    disable_event:<system>:<event>
                    stacktrace
                    snapshot
                    dump
                    cpudump
           example: echo do_fault:traceoff > set_ftrace_filter
                    echo do_trap:traceoff:3 > set_ftrace_filter
           The first one will disable tracing every time do_fault is hit
           The second will disable tracing at most 3 times when do_trap is hit
             The first time do trap is hit and it disables tracing, the
             counter will decrement to 2. If tracing is already disabled,
             the counter will not decrement. It only decrements when the
             trigger did work
           To remove trigger without count:
             echo '!<function>:<trigger> > set_ftrace_filter
           To remove trigger with a count:
             echo '!<function>:<trigger>:0 > set_ftrace_filter
set_ftrace_notrace    - echo function name in here to never trace.
          accepts: func_full_name, *func_end, func_begin*, *func_middle*
          modules: Can select a group via module command :mod:
          Does not accept triggers
```



```
$ sudo ./ping_ftrace.sh
[+] resetting ftrace
trace-cmd reset    (patience, pl...)
resetting set_ftrace_filter
resetting set_ftrace_notrace
resetting set_ftrace_notrace_pid
resetting set_ftrace_pid
resetting trace_options to defaults (as of 5.10.60)
resetting options/funcgraph-*
running '/usr/sbin/reset-ftrace-perf -q' now...
[+] tracer : function_graph
[+] setting options
[+] setting buffer size to 82 MB / cpu
[+] Function filtering:
 Regular filtering (via available_filter_functions):
 Setting filters for networking funcs only...
[+] filter: remove unwanted functions          (patience, pl...)
# of functions now being traced: 6649
[+] module filtering (for e1000)
e1000                 143360  0
[+] setting filter command: :mod:e1000
[+] Setting up wrapper runner process now...
[+] Tracing PID 1556 on CPU 1 now ...
> runner:1556: triggered
PING packtpub.com (104.22.1.175) 56(84) bytes of data.
64 bytes from 104.22.1.175 (104.22.1.175): icmp_seq=1 ttl=63 time=15.2 ms

--- packtpub.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 15.167/15.167/15.167/0.000 ms
Ftrace report:
-rw-r--r-- 1 root root 272K Feb 25 13:23 /home/letsdebug/Linux-Kernel-Debugging/ch9/ftrace/ftrace_reports/pin
g_ftrace.sh_20220225.txt
$
```



```
3024.237642 |   1)   ping-1869    |  .... |               |   sock_sendmsg() {
3024.237643 |   1)   ping-1869    |  .... |               |     security_socket_sendmsg() {
3024.237644 |   1)   ping-1869    |  .... |               |       apparmor_socket_sendmsg() {
3024.237645 |   1)   ping-1869    |  .... |   0.738 us    |         aa_sk_perm();
3024.237646 |   1)   ping-1869    |  .... |   2.587 us    |       } /* apparmor_socket_sendmsg */
3024.237647 |   1)   ping-1869    |  .... |   4.816 us    |     } /* security_socket_sendmsg */
3024.237649 |   1)   ping-1869    |  .... |               |     inet_sendmsg() {
3024.237650 |   1)   ping-1869    |  .... |               |       inet_send_prepare() {
3024.237652 |   1)   ping-1869    |  .... |               |         inet_autobind() {
3024.237652 |   1)   ping-1869    |  .... |   1.262 us    |           lock_sock_nested();
3024.237655 |   1)   ping-1869    |  .... |   1.135 us    |           _raw_write_lock_bh();
3024.237658 |   1)   ping-1869    |  ...1 |   1.537 us    |           sock_prot_inuse_add();
3024.237661 |   1)   ping-1869    |  ...1 |   1.115 us    |           _raw_write_unlock_bh();
3024.237663 |   1)   ping-1869    |  .... |   1.475 us    |           release_sock();
3024.237665 |   1)   ping-1869    |  .... | + 13.470 us   |         } /* inet_autobind */
3024.237666 |   1)   ping-1869    |  .... | + 15.571 us   |       } /* inet_send_prepare */
```

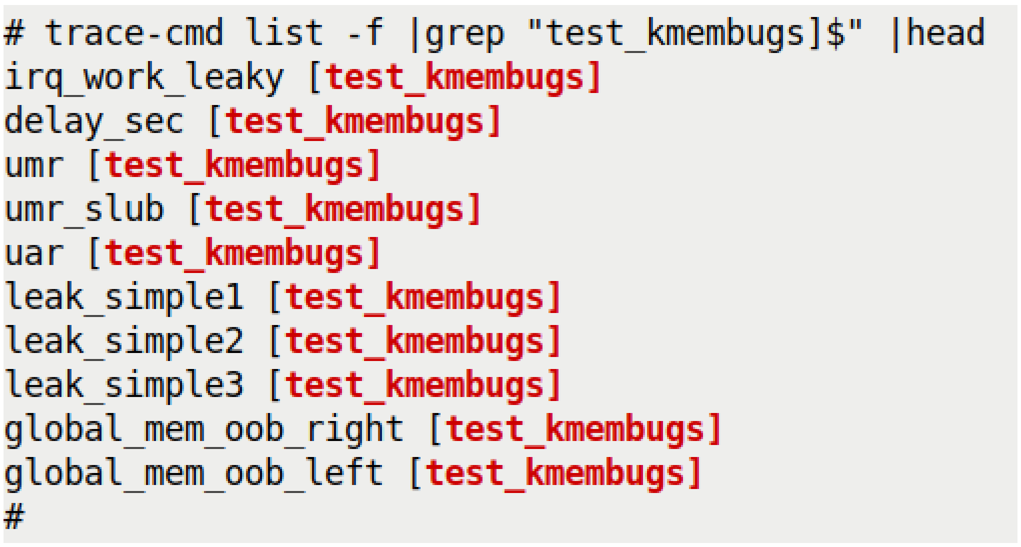


```
# pwd
/sys/kernel/tracing
# ls events/
alarmtimer/     ftrace/         mce/             random/         task/
avc/            gpio/           mdio/            ras/            tcp/
block/          header_event    migrate/         raw_syscalls/   thermal/
bpf_test_run/   header_page     mmap/            rcu/            thermal_power_allocator/
bpf_trace/      huge_memory/    mmc/             regmap/         timer/
cgroup/         hwmon/          module/          regulator/      tlb/
clk/            i2c/            msr/             resctrl/        udp/
compaction/     initcall/       napi/            rpm/            vmscan/
cpuhp/          intel_iommu/    neigh/           rseq/           vsyscall/
devfreq/        interconnect/   net/             rtc/            wbt/
dma_fence/      iocost/         nmi/             sched/          workqueue/
drm/            iomap/          oom/             scsi/           writeback/
enable          iommu/          page_isolation/  signal/         x86_fpu/
exceptions/     io_uring/       pagemap/         skb/            xdp/
ext4/           irq/            page_pool/       smbus/          xen/
fib/            irq_matrix/     percpu/          sock/           xhci-hcd/
fib6/           irq_vectors/    power/           spi/
filelock/       jbd2/           printk/          swiotlb/
filemap/        kmem/           pwm/             sync_trace/
fs_dax/         libata/         qdisc/           syscalls/
#
```

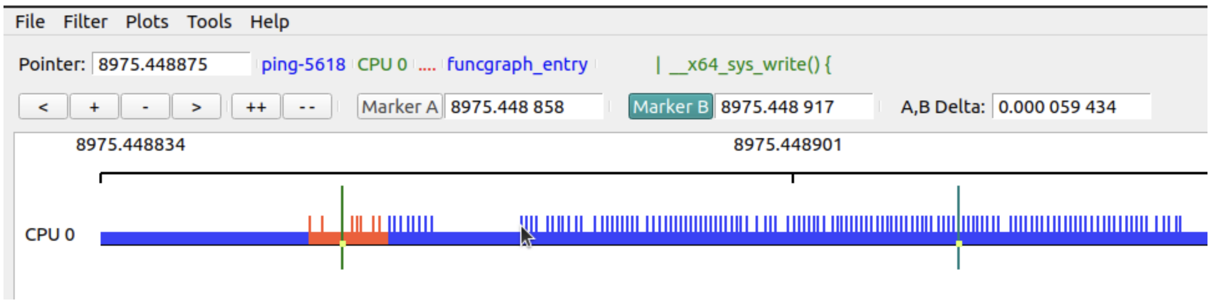
```
ftrace_dump_on_oops[=orig_cpu]
                [FTRACE] will dump the trace buffers on oops.
                If no parameter is passed, ftrace will dump
                buffers of all CPUs, but if you pass orig_cpu, it will
                dump only the buffer of the CPU that triggered the
                oops.
```

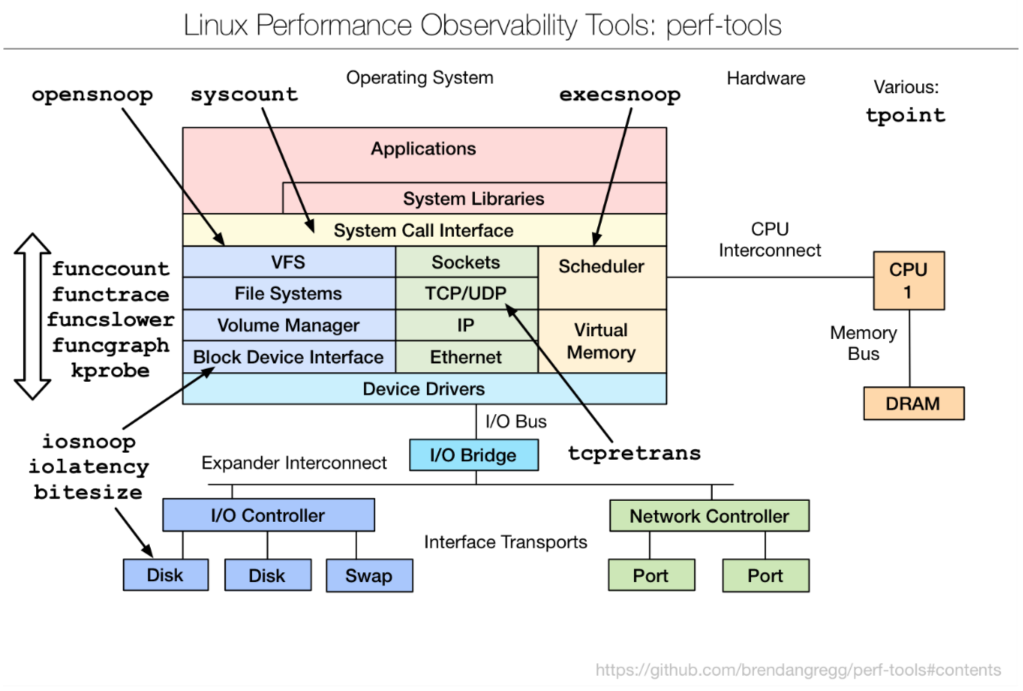
```
# ~/lkdsrc/ch9/ftrace/ftrc_1s.sh
trace-cmd reset
resetting set_ftrace_filter
resetting set_ftrace_notrace
resetting set_ftrace_notrace_pid
resetting set_ftrace_pid
resetting trace_options to defaults (as of 5.10.60)
resetting options/funcgraph-*
running '/usr/sbin/reset-ftrace-perf -q' now...
Tracing with function_graph for 1s ...
-rw-r--r-- 1 root root 371K Jan 28 12:40 /root/ftrace_reports/ftrc_1s.sh_20220128_124002.txt
#
# cat stack_max_size
4224
#
# cat stack_trace
        Depth    Size   Location    (35 entries)
        -----    ----   --------
  0)     4280      64   decay_load+0x5/0xa0
  1)     4216      96   __update_load_avg_se+0x22b/0x2c0
  2)     4120      88   update_load_avg+0x2c9/0x6f0
  3)     4032     136   update_blocked_averages+0x4c5/0x6a0
  4)     3896      24   update_nohz_stats+0x44/0x60
  5)     3872     296   update_sd_lb_stats.constprop.0+0x433/0xff0
  6)     3576     256   find_busiest_group+0x4d/0x370
  7)     3320     336   load_balance+0x168/0x1630
  8)     2984      96   newidle_balance+0x31a/0x470
  9)     2888      72   pick_next_task_fair+0x41/0x470
 10)     2816     128   __schedule+0x32e/0xc90
 11)     2688      32   schedule+0x4e/0xf0
 12)     2656      24   io_schedule+0x16/0x40
```

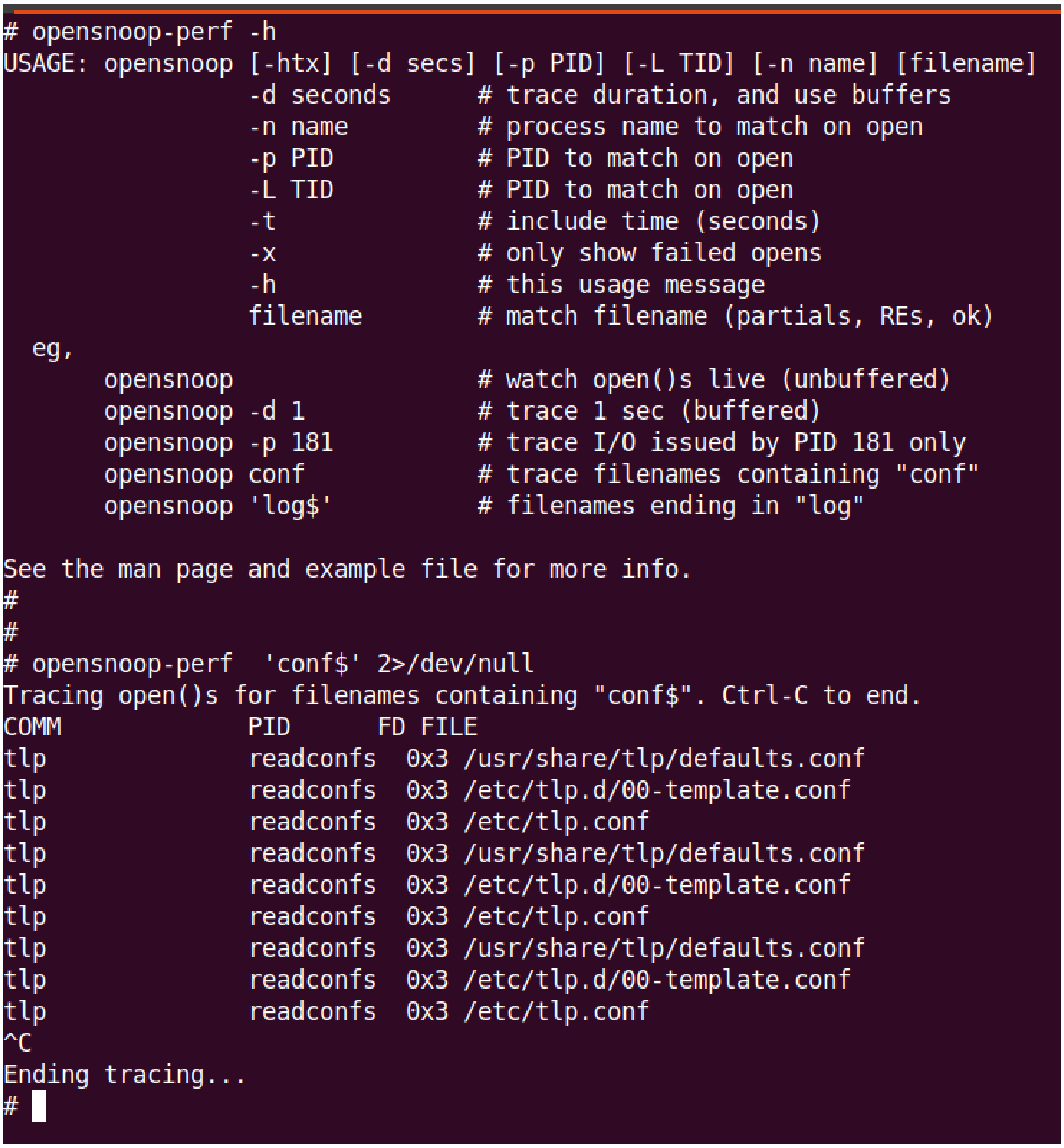
```
$ trace-cmd

trace-cmd version 2.8.3

usage:
  trace-cmd [COMMAND] ...

  commands:
     record - record a trace into a trace.dat file
     start - start tracing without recording into a file
     extract - extract a trace from the kernel
     stop - stop the kernel from recording trace data
     restart - restart the kernel trace data recording
     show - show the contents of the kernel tracing buffer
     reset - disable all kernel tracing and clear the trace buffers
     clear - clear the trace buffers
     report - read out the trace stored in a trace.dat file
     stream - Start tracing and read the output directly
     profile - Start profiling and read the output directly
     hist - show a histogram of the trace.dat information
     stat - show the status of the running tracing (ftrace) system
     split - parse a trace.dat file into smaller file(s)
     options - list the plugin options available for trace-cmd report
     listen - listen on a network socket for trace clients
     list - list the available events, plugins or options
     restore - restore a crashed record
     snapshot - take snapshot of running trace
     stack - output, enable or disable kernel stack tracing
     check-events - parse trace event formats

$ man trace-cmd-
trace-cmd-check-events  trace-cmd-profile       trace-cmd-split
trace-cmd-extract       trace-cmd-record        trace-cmd-stack
trace-cmd-hist          trace-cmd-report        trace-cmd-start
trace-cmd-list          trace-cmd-reset         trace-cmd-stat
trace-cmd-listen        trace-cmd-restore       trace-cmd-stop
trace-cmd-mem           trace-cmd-show          trace-cmd-stream
trace-cmd-options       trace-cmd-snapshot
```



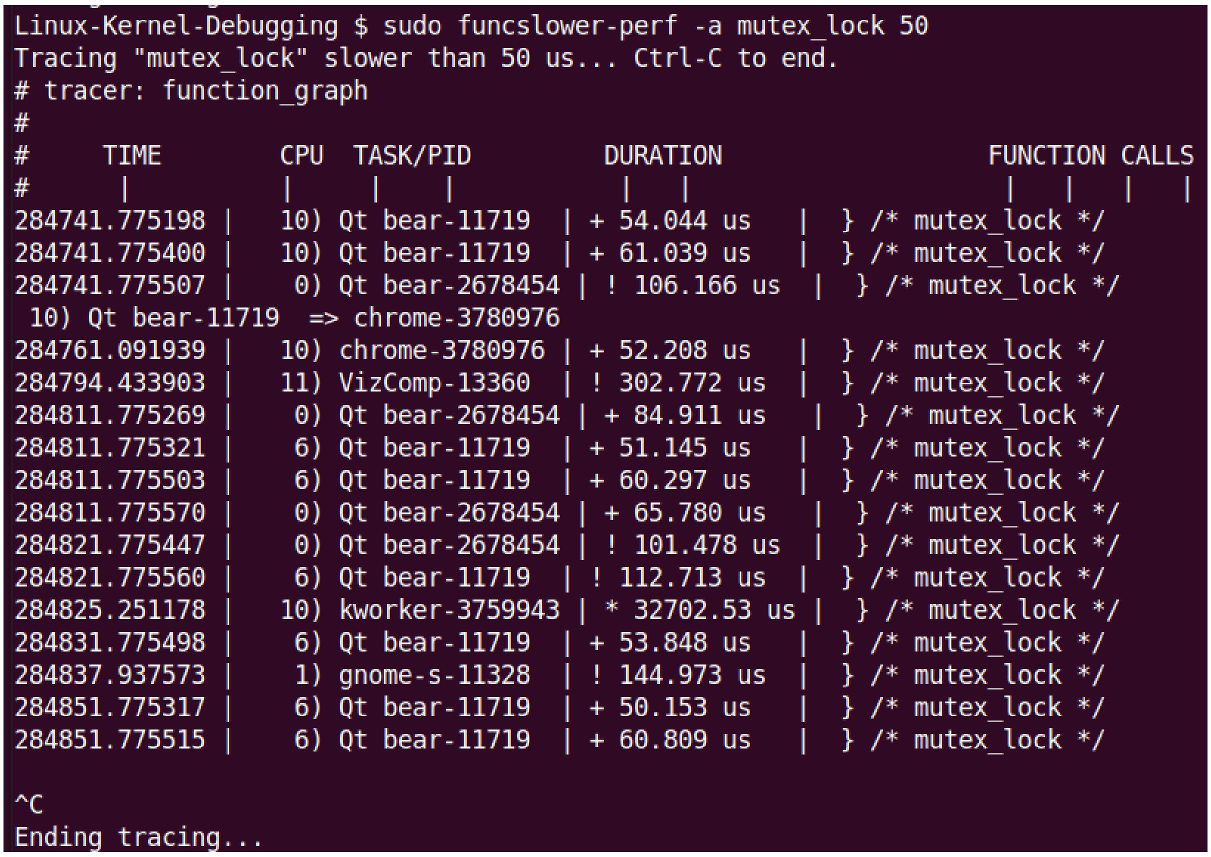
```
# trace-cmd list -f |grep "test_kmembugs]$" |head
irq_work_leaky [test_kmembugs]
delay_sec [test_kmembugs]
umr [test_kmembugs]
umr_slub [test_kmembugs]
uar [test_kmembugs]
leak_simple1 [test_kmembugs]
leak_simple2 [test_kmembugs]
leak_simple3 [test_kmembugs]
global_mem_oob_right [test_kmembugs]
global_mem_oob_left [test_kmembugs]
#
```



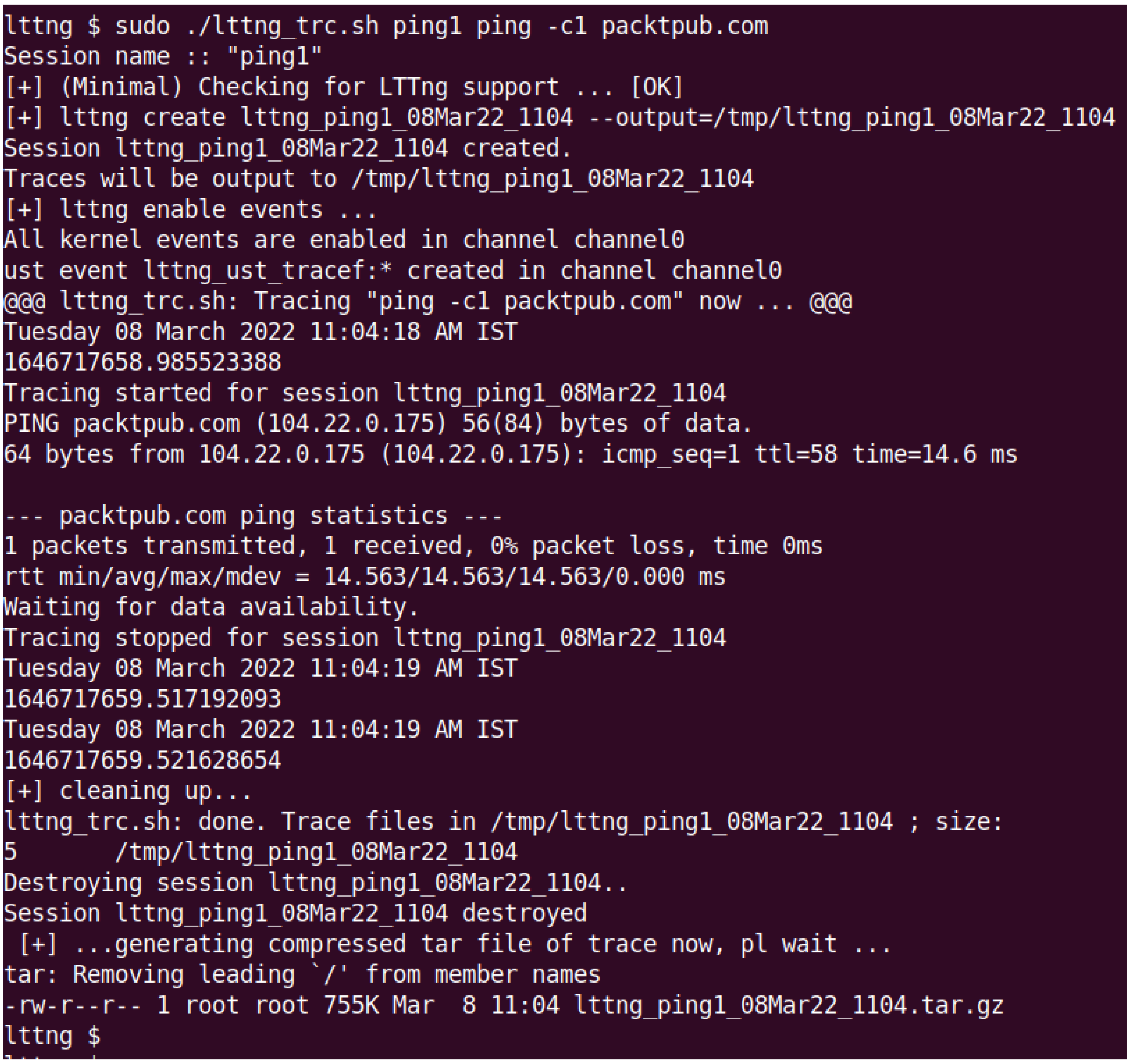




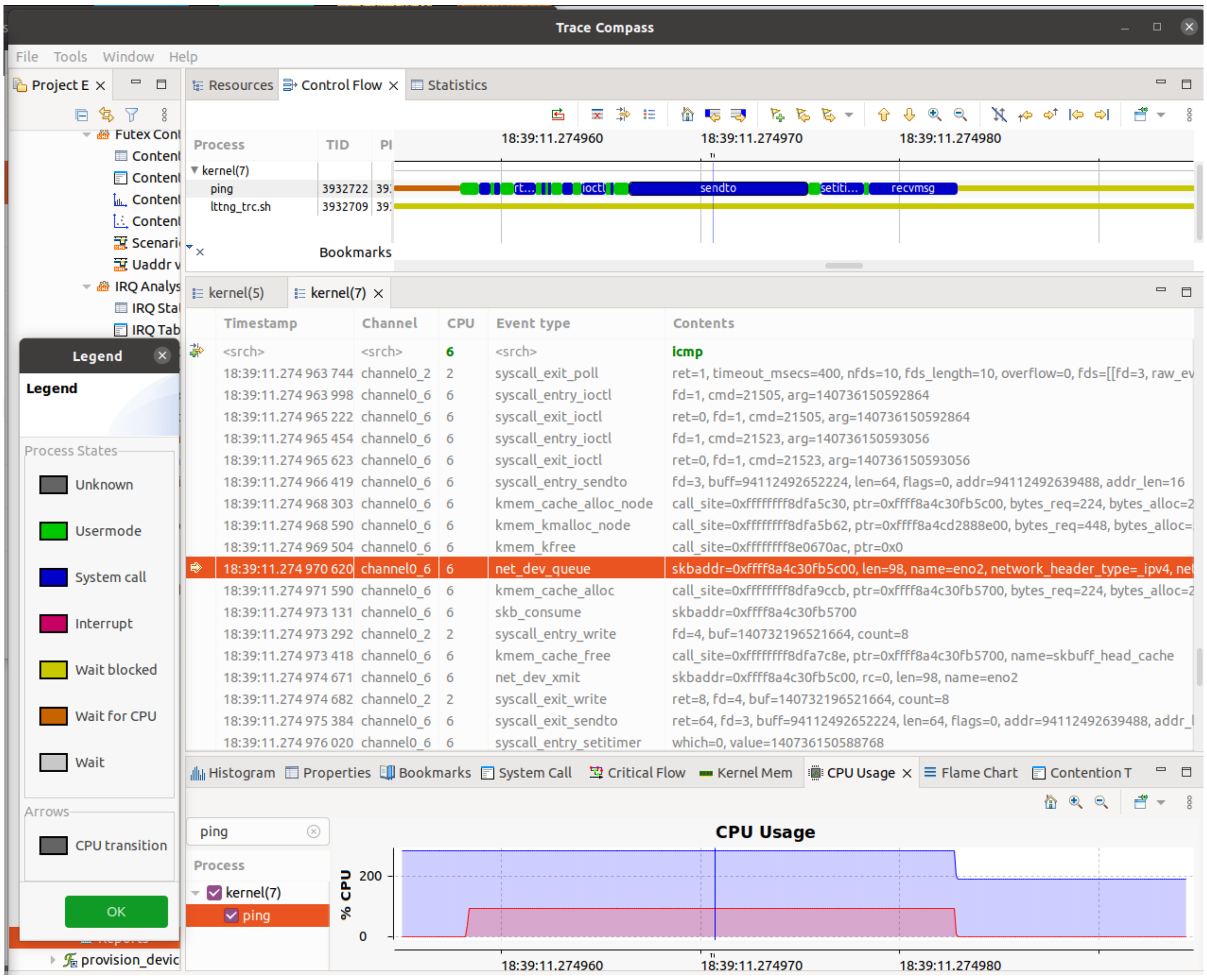
```
# opensnoop-perf -h
USAGE: opensnoop [-htx] [-d secs] [-p PID] [-L TID] [-n name] [filename]
                 -d seconds      # trace duration, and use buffers
                 -n name         # process name to match on open
                 -p PID          # PID to match on open
                 -L TID          # PID to match on open
                 -t              # include time (seconds)
                 -x              # only show failed opens
                 -h              # this usage message
                 filename        # match filename (partials, REs, ok)
  eg,
       opensnoop                 # watch open()s live (unbuffered)
       opensnoop -d 1            # trace 1 sec (buffered)
       opensnoop -p 181          # trace I/O issued by PID 181 only
       opensnoop conf            # trace filenames containing "conf"
       opensnoop 'log$'          # filenames ending in "log"

See the man page and example file for more info.
#
#
# opensnoop-perf  'conf$' 2>/dev/null
Tracing open()s for filenames containing "conf$". Ctrl-C to end.
COMM             PID      FD FILE
tlp              readconfs  0x3 /usr/share/tlp/defaults.conf
tlp              readconfs  0x3 /etc/tlp.d/00-template.conf
tlp              readconfs  0x3 /etc/tlp.conf
tlp              readconfs  0x3 /usr/share/tlp/defaults.conf
tlp              readconfs  0x3 /etc/tlp.d/00-template.conf
tlp              readconfs  0x3 /etc/tlp.conf
tlp              readconfs  0x3 /usr/share/tlp/defaults.conf
tlp              readconfs  0x3 /etc/tlp.d/00-template.conf
tlp              readconfs  0x3 /etc/tlp.conf
^C
Ending tracing...
# 
```



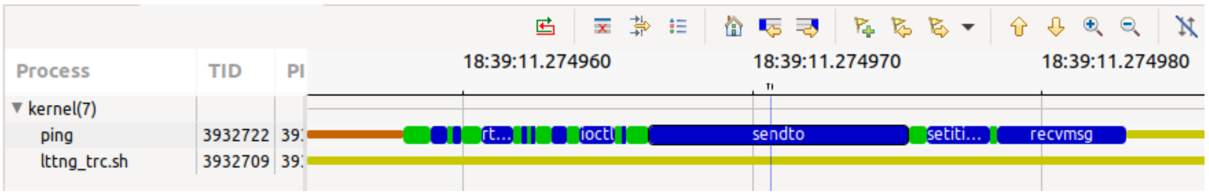
```
Linux-Kernel-Debugging $ sudo funcslower-perf -a mutex_lock 50
Tracing "mutex_lock" slower than 50 us... Ctrl-C to end.
# tracer: function_graph
#
#     TIME        CPU  TASK/PID         DURATION                  FUNCTION CALLS
#      |           |     |    |           |   |                     |   |   |   |
284741.775198 |   10) Qt bear-11719  | + 54.044 us   |  } /* mutex_lock */
284741.775400 |   10) Qt bear-11719  | + 61.039 us   |  } /* mutex_lock */
284741.775507 |    0) Qt bear-2678454 | ! 106.166 us  |  } /* mutex_lock */
 10) Qt bear-11719  => chrome-3780976
284761.091939 |   10) chrome-3780976 | + 52.208 us   |  } /* mutex_lock */
284794.433903 |   11) VizComp-13360  | ! 302.772 us  |  } /* mutex_lock */
284811.775269 |    0) Qt bear-2678454 | + 84.911 us   |  } /* mutex_lock */
284811.775321 |    6) Qt bear-11719  | + 51.145 us   |  } /* mutex_lock */
284811.775503 |    6) Qt bear-11719  | + 60.297 us   |  } /* mutex_lock */
284811.775570 |    0) Qt bear-2678454 | + 65.780 us   |  } /* mutex_lock */
284821.775447 |    0) Qt bear-2678454 | ! 101.478 us  |  } /* mutex_lock */
284821.775560 |    6) Qt bear-11719  | ! 112.713 us  |  } /* mutex_lock */
284825.251178 |   10) kworker-3759943 | * 32702.53 us |  } /* mutex_lock */
284831.775498 |    6) Qt bear-11719  | + 53.848 us   |  } /* mutex_lock */
284837.937573 |    1) gnome-s-11328  | ! 144.973 us  |  } /* mutex_lock */
284851.775317 |    6) Qt bear-11719  | + 50.153 us   |  } /* mutex_lock */
284851.775515 |    6) Qt bear-11719  | + 60.809 us   |  } /* mutex_lock */

^C
Ending tracing...
```



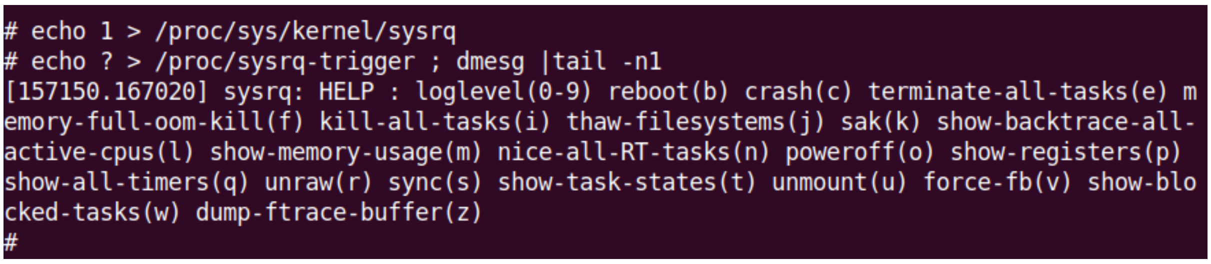
```
lttng $ sudo ./lttng_trc.sh ping1 ping -c1 packtpub.com
Session name :: "ping1"
[+] (Minimal) Checking for LTTng support ... [OK]
[+] lttng create lttng_ping1_08Mar22_1104 --output=/tmp/lttng_ping1_08Mar22_1104
Session lttng_ping1_08Mar22_1104 created.
Traces will be output to /tmp/lttng_ping1_08Mar22_1104
[+] lttng enable events ...
All kernel events are enabled in channel channel0
ust event lttng_ust_tracef:* created in channel channel0
@@@ lttng_trc.sh: Tracing "ping -c1 packtpub.com" now ... @@@
Tuesday 08 March 2022 11:04:18 AM IST
1646717658.985523388
Tracing started for session lttng_ping1_08Mar22_1104
PING packtpub.com (104.22.0.175) 56(84) bytes of data.
64 bytes from 104.22.0.175 (104.22.0.175): icmp_seq=1 ttl=58 time=14.6 ms

--- packtpub.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 14.563/14.563/14.563/0.000 ms
Waiting for data availability.
Tracing stopped for session lttng_ping1_08Mar22_1104
Tuesday 08 March 2022 11:04:19 AM IST
1646717659.517192093
Tuesday 08 March 2022 11:04:19 AM IST
1646717659.521628654
[+] cleaning up...
lttng_trc.sh: done. Trace files in /tmp/lttng_ping1_08Mar22_1104 ; size:
5       /tmp/lttng_ping1_08Mar22_1104
Destroying session lttng_ping1_08Mar22_1104..
Session lttng_ping1_08Mar22_1104 destroyed
 [+] ...generating compressed tar file of trace now, pl wait ...
tar: Removing leading `/' from member names
-rw-r--r-- 1 root root 755K Mar  8 11:04 lttng_ping1_08Mar22_1104.tar.gz
lttng $
```



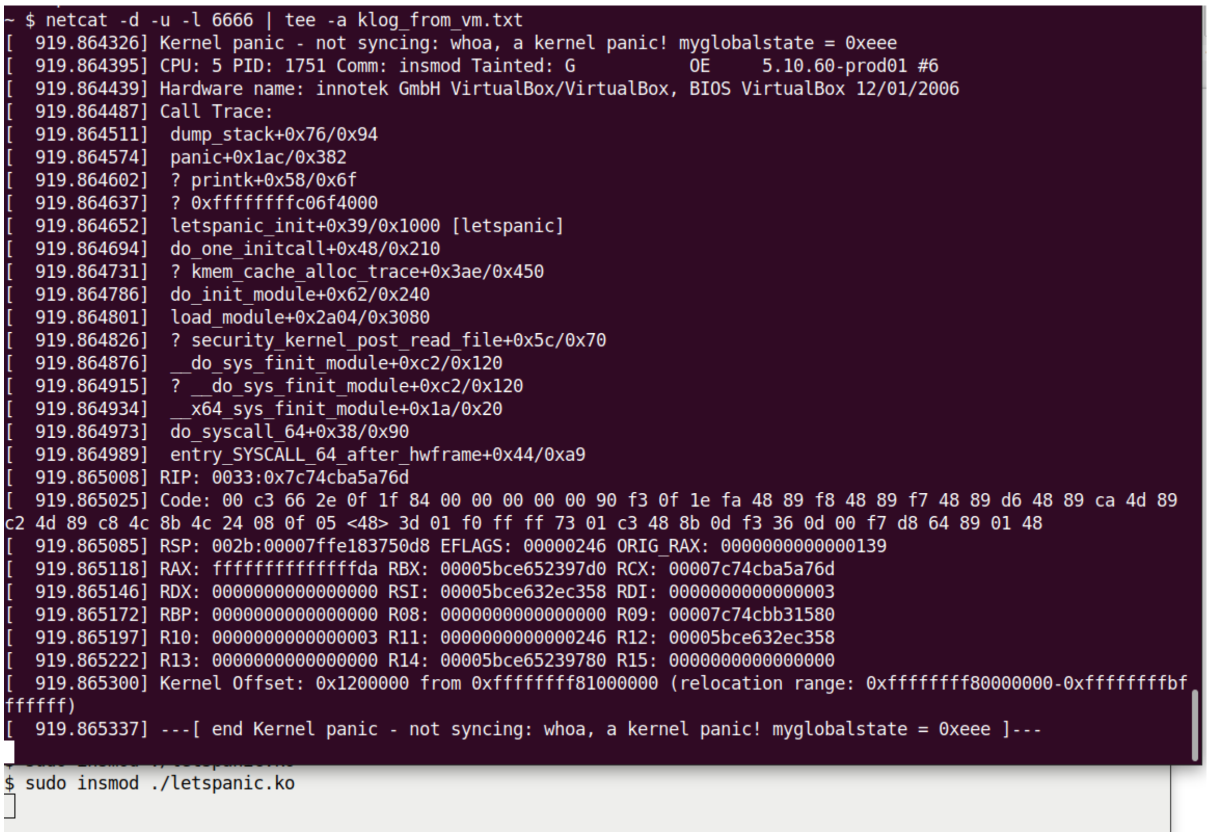
	Timestamp	Channel	CPU	Event type	Contents
	<srch>	<srch>	6	<srch>	icmp
	18:39:11.274 963 744	channel0_2	2	syscall_exit_poll	ret=1, timeout_msecs=400, nfds=10, fds_length=10, overflow=0, fds=[[fd=3, raw_ev
	18:39:11.274 963 998	channel0_6	6	syscall_entry_ioctl	fd=1, cmd=21505, arg=140736150592864
	18:39:11.274 965 222	channel0_6	6	syscall_exit_ioctl	ret=0, fd=1, cmd=21505, arg=140736150592864
	18:39:11.274 965 454	channel0_6	6	syscall_entry_ioctl	fd=1, cmd=21523, arg=140736150593056
	18:39:11.274 965 623	channel0_6	6	syscall_exit_ioctl	ret=0, fd=1, cmd=21523, arg=140736150593056
	18:39:11.274 966 419	channel0_6	6	syscall_entry_sendto	fd=3, buff=94112492652224, len=64, flags=0, addr=94112492639488, addr_len=16
	18:39:11.274 968 303	channel0_6	6	kmem_cache_alloc_node	call_site=0xffffffff8dfa5c30, ptr=0xffff8a4c30fb5c00, bytes_req=224, bytes_alloc=2
	18:39:11.274 968 590	channel0_6	6	kmem_kmalloc_node	call_site=0xffffffff8dfa5b62, ptr=0xffff8a4cd2888e00, bytes_req=448, bytes_alloc=
	18:39:11.274 969 504	channel0_6	6	kmem_kfree	call_site=0xffffffff8e0670ac, ptr=0x0
⇨	18:39:11.274 970 620	channel0_6	6	net_dev_queue	skbaddr=0xffff8a4c30fb5c00, len=98, name=eno2, network_header_type=_ipv4, net
	18:39:11.274 971 590	channel0_6	6	kmem_cache_alloc	call_site=0xffffffff8dfa9ccb, ptr=0xffff8a4c30fb5700, bytes_req=224, bytes_alloc=2
	18:39:11.274 973 131	channel0_6	6	skb_consume	skbaddr=0xffff8a4c30fb5700
	18:39:11.274 973 292	channel0_2	2	syscall_entry_write	fd=4, buf=140732196521664, count=8
	18:39:11.274 973 418	channel0_6	6	kmem_cache_free	call_site=0xffffffff8dfa7c8e, ptr=0xffff8a4c30fb5700, name=skbuff_head_cache
	18:39:11.274 974 671	channel0_6	6	net_dev_xmit	skbaddr=0xffff8a4c30fb5c00, rc=0, len=98, name=eno2
	18:39:11.274 974 682	channel0_2	2	syscall_exit_write	ret=8, fd=4, buf=140732196521664, count=8
	18:39:11.274 975 384	channel0_6	6	syscall_exit_sendto	ret=64, fd=3, buff=94112492652224, len=64, flags=0, addr=94112492639488, addr_l
	18:39:11.274 976 020	channel0_6	6	syscall_entry_setitimer	which=0, value=140736150588768



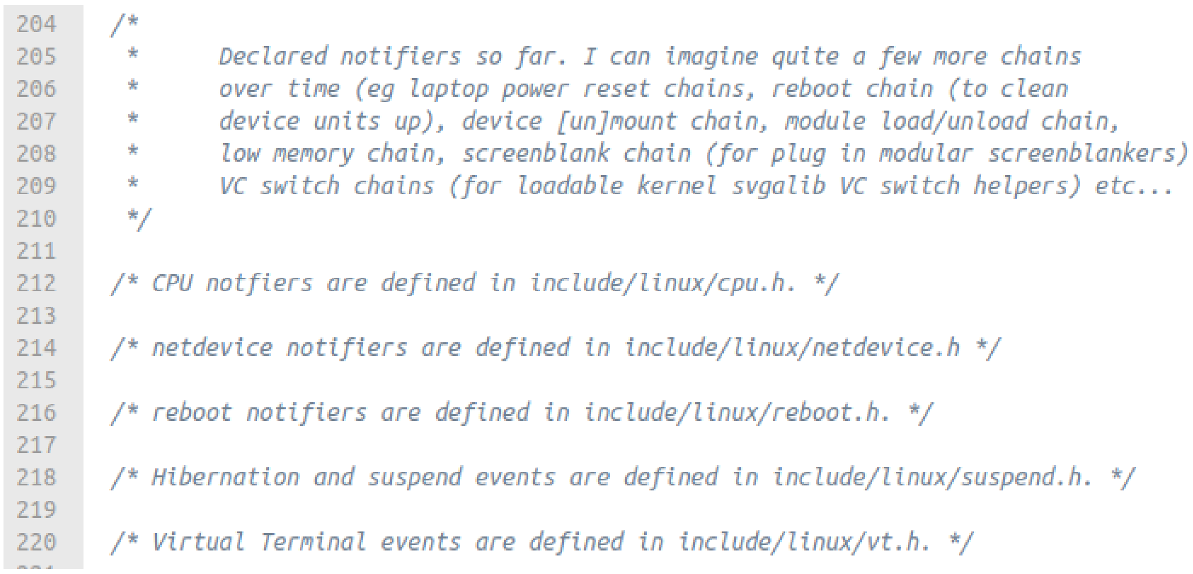
Chapter 10: Kernel Panic, Lockups, and Hangs



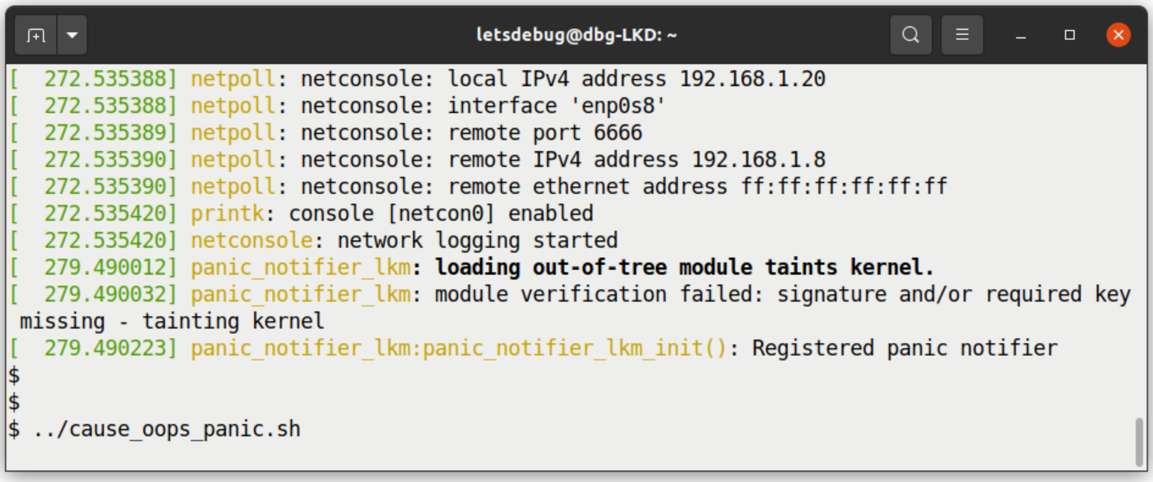
```
# echo 1 > /proc/sys/kernel/sysrq
# echo ? > /proc/sysrq-trigger ; dmesg |tail -n1
[157150.167020] sysrq: HELP : loglevel(0-9) reboot(b) crash(c) terminate-all-tasks(e) m
emory-full-oom-kill(f) kill-all-tasks(i) thaw-filesystems(j) sak(k) show-backtrace-all-
active-cpus(l) show-memory-usage(m) nice-all-RT-tasks(n) poweroff(o) show-registers(p)
show-all-timers(q) unraw(r) sync(s) show-task-states(t) unmount(u) force-fb(v) show-blo
cked-tasks(w) dump-ftrace-buffer(z)
#
```



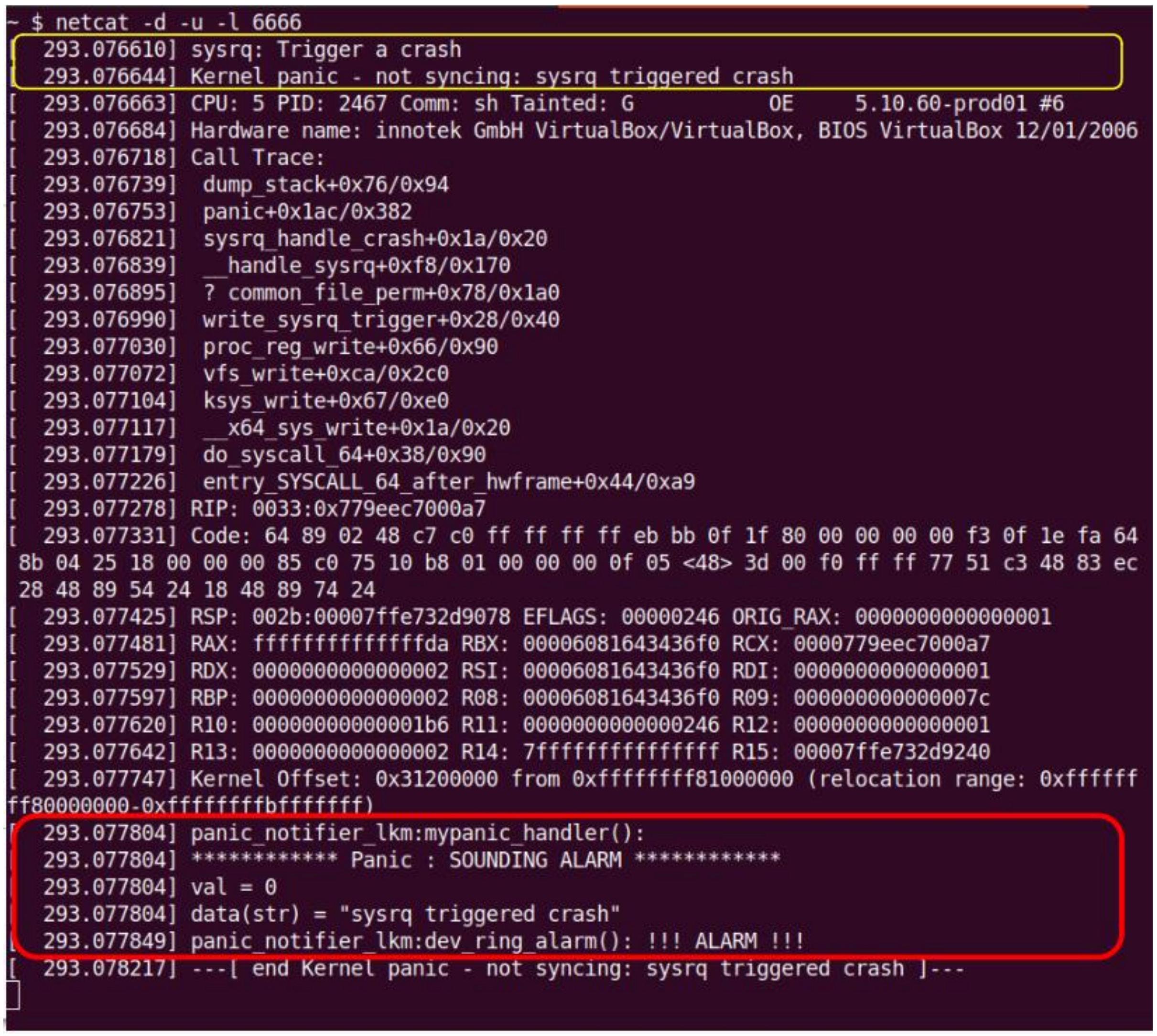
```
~ $ netcat -d -u -l 6666 | tee -a klog_from_vm.txt
[  919.864326] Kernel panic - not syncing: whoa, a kernel panic! myglobalstate = 0xeee
[  919.864395] CPU: 5 PID: 1751 Comm: insmod Tainted: G           OE     5.10.60-prod01 #6
[  919.864439] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[  919.864487] Call Trace:
[  919.864511]  dump_stack+0x76/0x94
[  919.864574]  panic+0x1ac/0x382
[  919.864602]  ? printk+0x58/0x6f
[  919.864637]  ? 0xffffffffc06f4000
[  919.864652]  letspanic_init+0x39/0x1000 [letspanic]
[  919.864694]  do_one_initcall+0x48/0x210
[  919.864731]  ? kmem_cache_alloc_trace+0x3ae/0x450
[  919.864786]  do_init_module+0x62/0x240
[  919.864801]  load_module+0x2a04/0x3080
[  919.864826]  ? security_kernel_post_read_file+0x5c/0x70
[  919.864876]  __do_sys_finit_module+0xc2/0x120
[  919.864915]  ? __do_sys_finit_module+0xc2/0x120
[  919.864934]  __x64_sys_finit_module+0x1a/0x20
[  919.864973]  do_syscall_64+0x38/0x90
[  919.864989]  entry_SYSCALL_64_after_hwframe+0x44/0xa9
[  919.865008] RIP: 0033:0x7c74cba5a76d
[  919.865025] Code: 00 c3 66 2e 0f 1f 84 00 00 00 00 00 90 f3 0f 1e fa 48 89 f8 48 89 f7 48 89 d6 48 89 ca 4d 89
c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff ff 73 01 c3 48 8b 0d f3 36 0d 00 f7 d8 64 89 01 48
[  919.865085] RSP: 002b:00007ffe183750d8 EFLAGS: 00000246 ORIG_RAX: 0000000000000139
[  919.865118] RAX: ffffffffffffffda RBX: 00005bce652397d0 RCX: 00007c74cba5a76d
[  919.865146] RDX: 0000000000000000 RSI: 00005bce632ec358 RDI: 0000000000000003
[  919.865172] RBP: 0000000000000000 R08: 0000000000000000 R09: 00007c74cbb31580
[  919.865197] R10: 0000000000000003 R11: 0000000000000246 R12: 00005bce632ec358
[  919.865222] R13: 0000000000000000 R14: 00005bce65239780 R15: 0000000000000000
[  919.865300] Kernel Offset: 0x1200000 from 0xffffffff81000000 (relocation range: 0xffffffff80000000-0xffffffffbf
ffffff)
[  919.865337] ---[ end Kernel panic - not syncing: whoa, a kernel panic! myglobalstate = 0xeee ]---
$ sudo insmod ./letspanic.ko
```



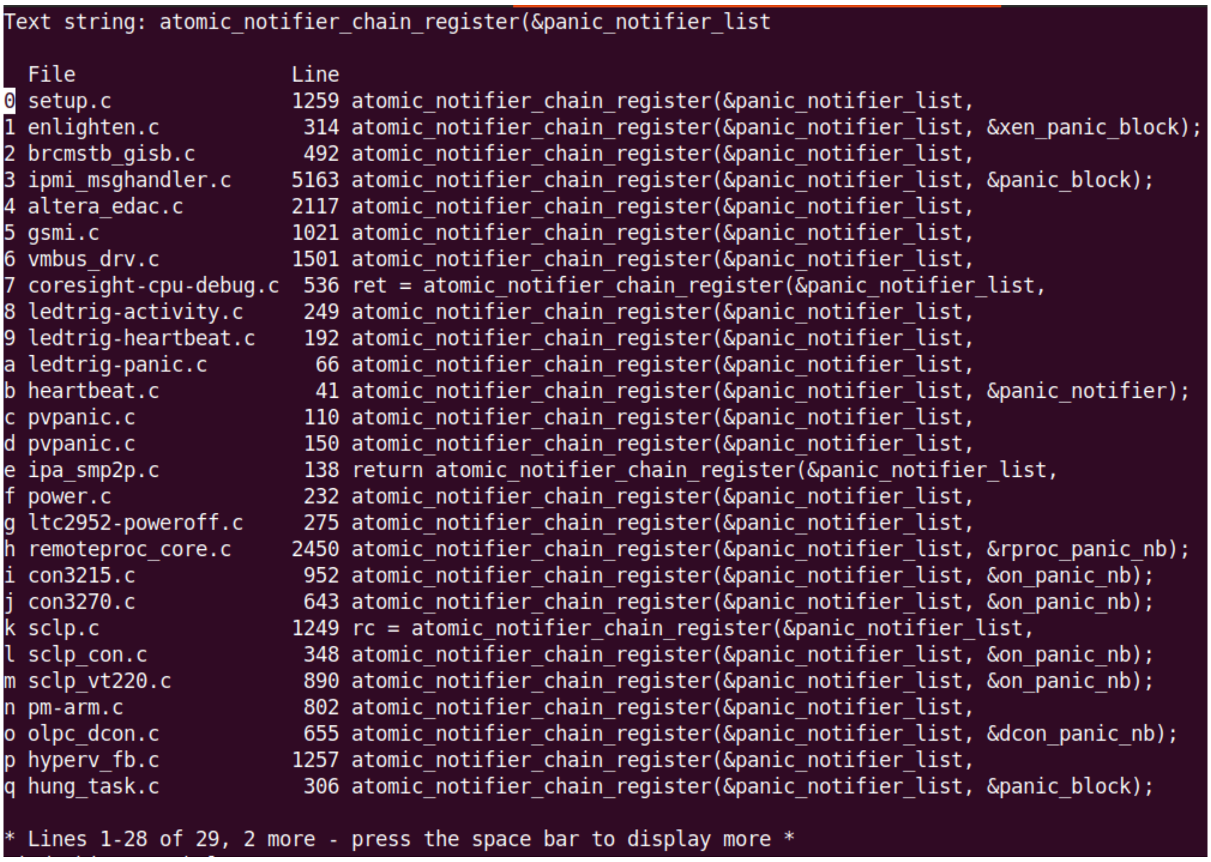
```
/*
 *      Declared notifiers so far. I can imagine quite a few more chains
 *      over time (eg laptop power reset chains, reboot chain (to clean
 *      device units up), device [un]mount chain, module load/unload chain,
 *      low memory chain, screenblank chain (for plug in modular screenblankers)
 *      VC switch chains (for loadable kernel svgalib VC switch helpers) etc...
 */

/* CPU notfiers are defined in include/linux/cpu.h. */

/* netdevice notifiers are defined in include/linux/netdevice.h */

/* reboot notifiers are defined in include/linux/reboot.h. */

/* Hibernation and suspend events are defined in include/linux/suspend.h. */

/* Virtual Terminal events are defined in include/linux/vt.h. */
```



```
[  272.535388] netpoll: netconsole: local IPv4 address 192.168.1.20
[  272.535388] netpoll: netconsole: interface 'enp0s8'
[  272.535389] netpoll: netconsole: remote port 6666
[  272.535390] netpoll: netconsole: remote IPv4 address 192.168.1.8
[  272.535390] netpoll: netconsole: remote ethernet address ff:ff:ff:ff:ff:ff
[  272.535420] printk: console [netcon0] enabled
[  272.535420] netconsole: network logging started
[  279.490012] panic_notifier_lkm: loading out-of-tree module taints kernel.
[  279.490032] panic_notifier_lkm: module verification failed: signature and/or required key
 missing - tainting kernel
[  279.490223] panic_notifier_lkm:panic_notifier_lkm_init(): Registered panic notifier
$
$
$ ../cause_oops_panic.sh
```

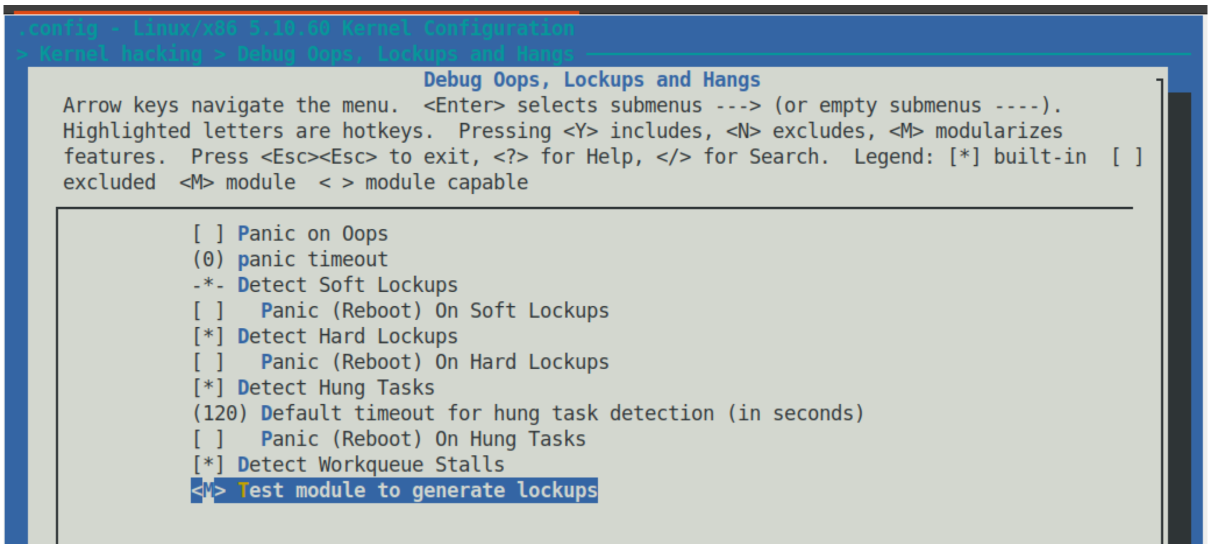


```
~ $ netcat -d -u -l 6666
[  293.076610] sysrq: Trigger a crash
[  293.076644] Kernel panic - not syncing: sysrq triggered crash
[  293.076663] CPU: 5 PID: 2467 Comm: sh Tainted: G           OE     5.10.60-prod01 #6
[  293.076684] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[  293.076718] Call Trace:
[  293.076739]  dump_stack+0x76/0x94
[  293.076753]  panic+0x1ac/0x382
[  293.076821]  sysrq_handle_crash+0x1a/0x20
[  293.076839]  __handle_sysrq+0xf8/0x170
[  293.076895]  ? common_file_perm+0x78/0x1a0
[  293.076990]  write_sysrq_trigger+0x28/0x40
[  293.077030]  proc_reg_write+0x66/0x90
[  293.077072]  vfs_write+0xca/0x2c0
[  293.077104]  ksys_write+0x67/0xe0
[  293.077117]  __x64_sys_write+0x1a/0x20
[  293.077179]  do_syscall_64+0x38/0x90
[  293.077226]  entry_SYSCALL_64_after_hwframe+0x44/0xa9
[  293.077278] RIP: 0033:0x779eec7000a7
[  293.077331] Code: 64 89 02 48 c7 c0 ff ff ff ff eb bb 0f 1f 80 00 00 00 00 f3 0f 1e fa 64
 8b 04 25 18 00 00 00 85 c0 75 10 b8 01 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 51 c3 48 83 ec
 28 48 89 54 24 18 48 89 74 24
[  293.077425] RSP: 002b:00007ffe732d9078 EFLAGS: 00000246 ORIG_RAX: 0000000000000001
[  293.077481] RAX: ffffffffffffffda RBX: 00006081643436f0 RCX: 0000779eec7000a7
[  293.077529] RDX: 0000000000000002 RSI: 00006081643436f0 RDI: 0000000000000001
[  293.077597] RBP: 0000000000000002 R08: 00006081643436f0 R09: 000000000000007c
[  293.077620] R10: 00000000000001b6 R11: 0000000000000246 R12: 0000000000000001
[  293.077642] R13: 0000000000000002 R14: 7fffffffffffffff R15: 00007ffe732d9240
[  293.077747] Kernel Offset: 0x31200000 from 0xffffffff81000000 (relocation range: 0xffffff
ff80000000-0xffffffffbfffffff)
[  293.077804] panic_notifier_lkm:mypanic_handler():
[  293.077804] ************ Panic : SOUNDING ALARM ************
[  293.077804] val = 0
[  293.077804] data(str) = "sysrq triggered crash"
[  293.077849] panic_notifier_lkm:dev_ring_alarm(): !!! ALARM !!!
[  293.078217] ---[ end Kernel panic - not syncing: sysrq triggered crash ]---
```

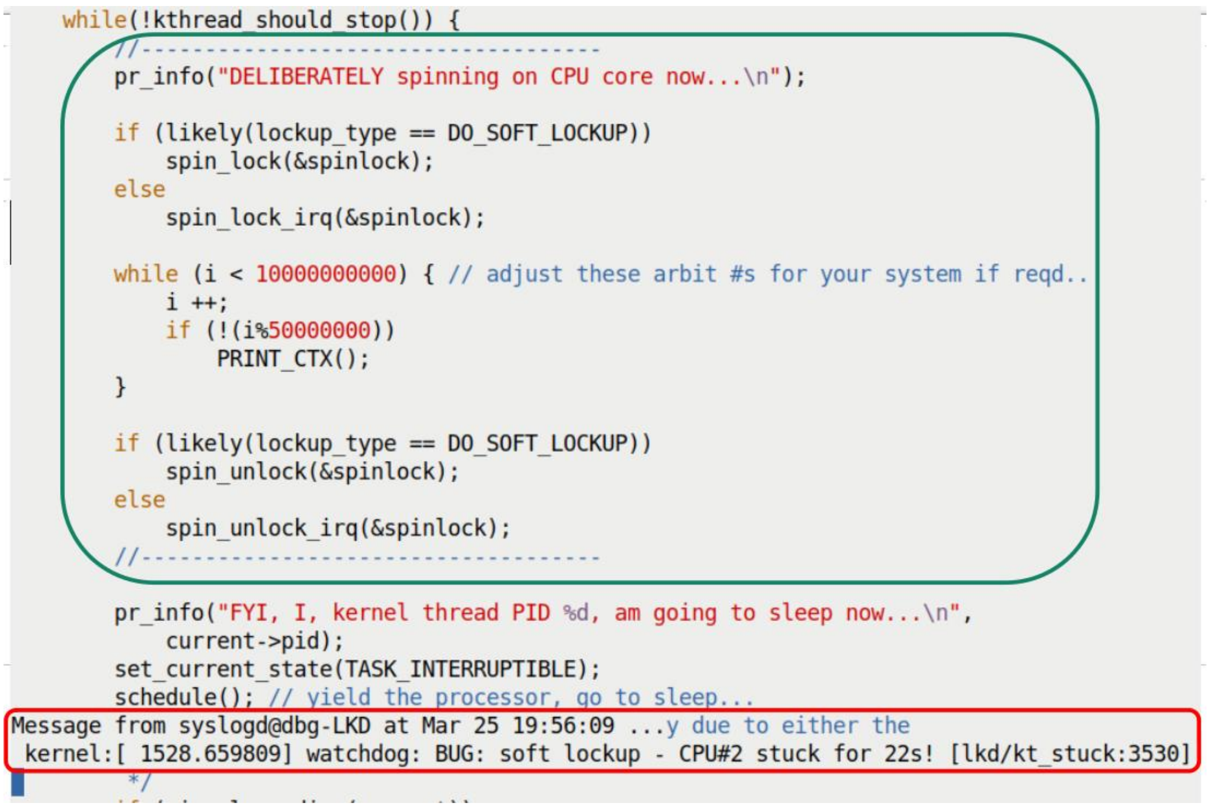


```
Text string: atomic_notifier_chain_register(&panic_notifier_list

  File                    Line
0 setup.c                 1259 atomic_notifier_chain_register(&panic_notifier_list,
1 enlighten.c              314 atomic_notifier_chain_register(&panic_notifier_list, &xen_panic_block);
2 brcmstb_gisb.c           492 atomic_notifier_chain_register(&panic_notifier_list,
3 ipmi_msghandler.c       5163 atomic_notifier_chain_register(&panic_notifier_list, &panic_block);
4 altera_edac.c           2117 atomic_notifier_chain_register(&panic_notifier_list,
5 gsmi.c                  1021 atomic_notifier_chain_register(&panic_notifier_list,
6 vmbus_drv.c             1501 atomic_notifier_chain_register(&panic_notifier_list,
7 coresight-cpu-debug.c    536 ret = atomic_notifier_chain_register(&panic_notifier_list,
8 ledtrig-activity.c       249 atomic_notifier_chain_register(&panic_notifier_list,
9 ledtrig-heartbeat.c      192 atomic_notifier_chain_register(&panic_notifier_list,
a ledtrig-panic.c           66 atomic_notifier_chain_register(&panic_notifier_list,
b heartbeat.c               41 atomic_notifier_chain_register(&panic_notifier_list, &panic_notifier);
c pvpanic.c                110 atomic_notifier_chain_register(&panic_notifier_list,
d pvpanic.c                150 atomic_notifier_chain_register(&panic_notifier_list,
e ipa_smp2p.c              138 return atomic_notifier_chain_register(&panic_notifier_list,
f power.c                  232 atomic_notifier_chain_register(&panic_notifier_list,
g ltc2952-poweroff.c       275 atomic_notifier_chain_register(&panic_notifier_list,
h remoteproc_core.c       2450 atomic_notifier_chain_register(&panic_notifier_list, &rproc_panic_nb);
i con3215.c                952 atomic_notifier_chain_register(&panic_notifier_list, &on_panic_nb);
j con3270.c                643 atomic_notifier_chain_register(&panic_notifier_list, &on_panic_nb);
k sclp.c                  1249 rc = atomic_notifier_chain_register(&panic_notifier_list,
l sclp_con.c               348 atomic_notifier_chain_register(&panic_notifier_list, &on_panic_nb);
m sclp_vt220.c             890 atomic_notifier_chain_register(&panic_notifier_list, &on_panic_nb);
n pm-arm.c                 802 atomic_notifier_chain_register(&panic_notifier_list,
o olpc_dcon.c              655 atomic_notifier_chain_register(&panic_notifier_list, &dcon_panic_nb);
p hyperv_fb.c             1257 atomic_notifier_chain_register(&panic_notifier_list,
q hung_task.c              306 atomic_notifier_chain_register(&panic_notifier_list, &panic_block);

* Lines 1-28 of 29, 2 more - press the space bar to display more *
```

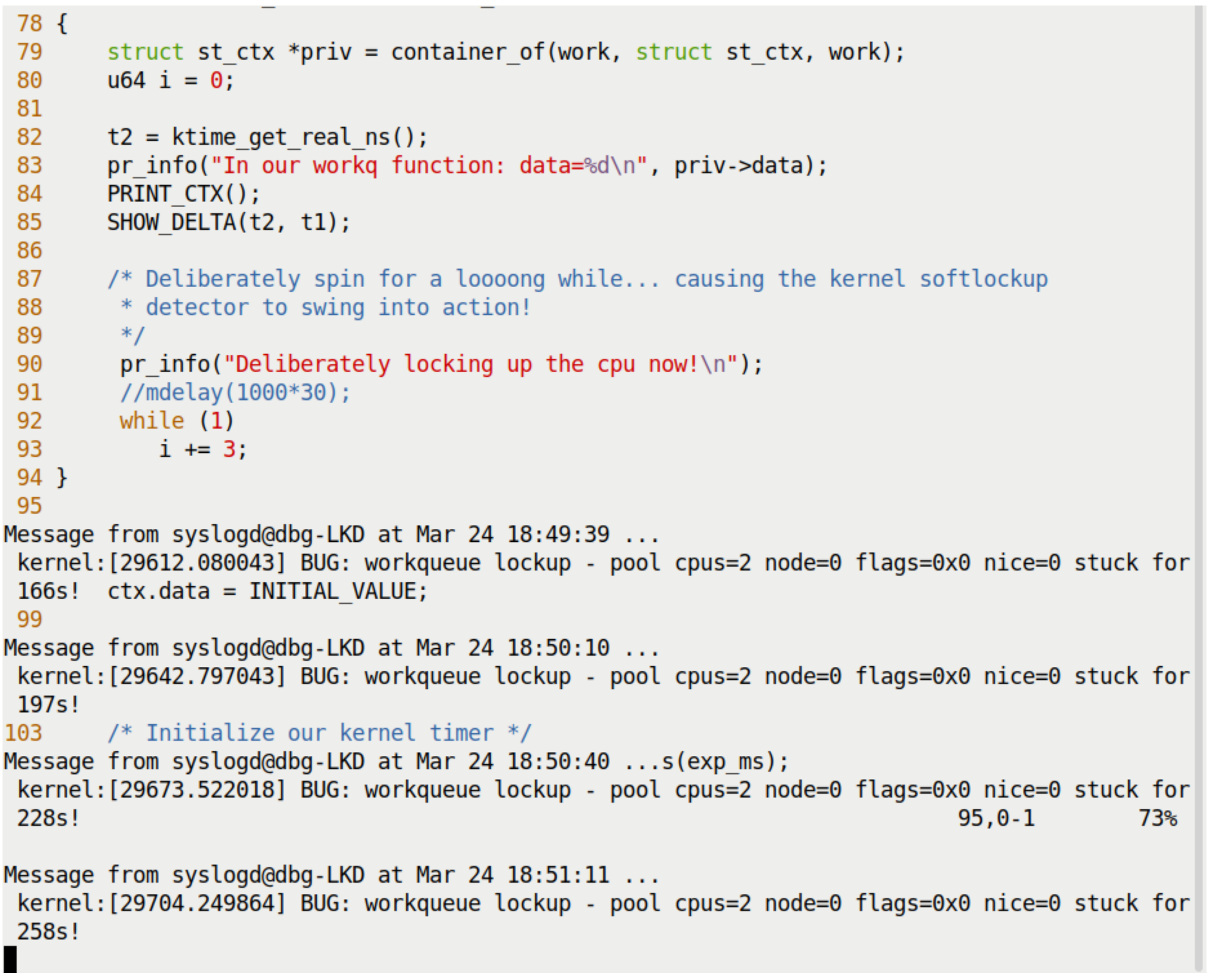
```
Integer 'retry-timeout' found = 60
Integer 'repair-maximum' found = 2
String 'watchdog-device' found as '/dev/watchdog'
Variable 'realtime' found as 'yes' = 1
Integer 'priority' found = 1
starting daemon (5.15)
int=1s realtime=yes sync=no load=0,0,0 soft=no
memory not checked
ping
file
pidfile
interface
temperature
no test binary files
no repair binary files
error retry time-out = 60 seconds
repair attempts = 2
alive=/dev/watchdog heartbeat=[none] to=root no_act=no force=no
watchdog now set to 60 seconds
hardware watchdog identity
still alive after 1 interval(s)
still alive after 2 interval(s)
still alive after 3 interval(s)
```



```
.config - Linux/x86 5.10.60 Kernel Configuration
> Kernel hacking > Debug Oops, Lockups and Hangs
                         Debug Oops, Lockups and Hangs
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes
  features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ]
  excluded  <M> module  < > module capable

            [ ] Panic on Oops
            (0) panic timeout
            -*- Detect Soft Lockups
            [ ]   Panic (Reboot) On Soft Lockups
            [*] Detect Hard Lockups
            [ ]   Panic (Reboot) On Hard Lockups
            [*] Detect Hung Tasks
            (120) Default timeout for hung task detection (in seconds)
            [ ]   Panic (Reboot) On Hung Tasks
            [*] Detect Workqueue Stalls
            <M> Test module to generate lockups
```

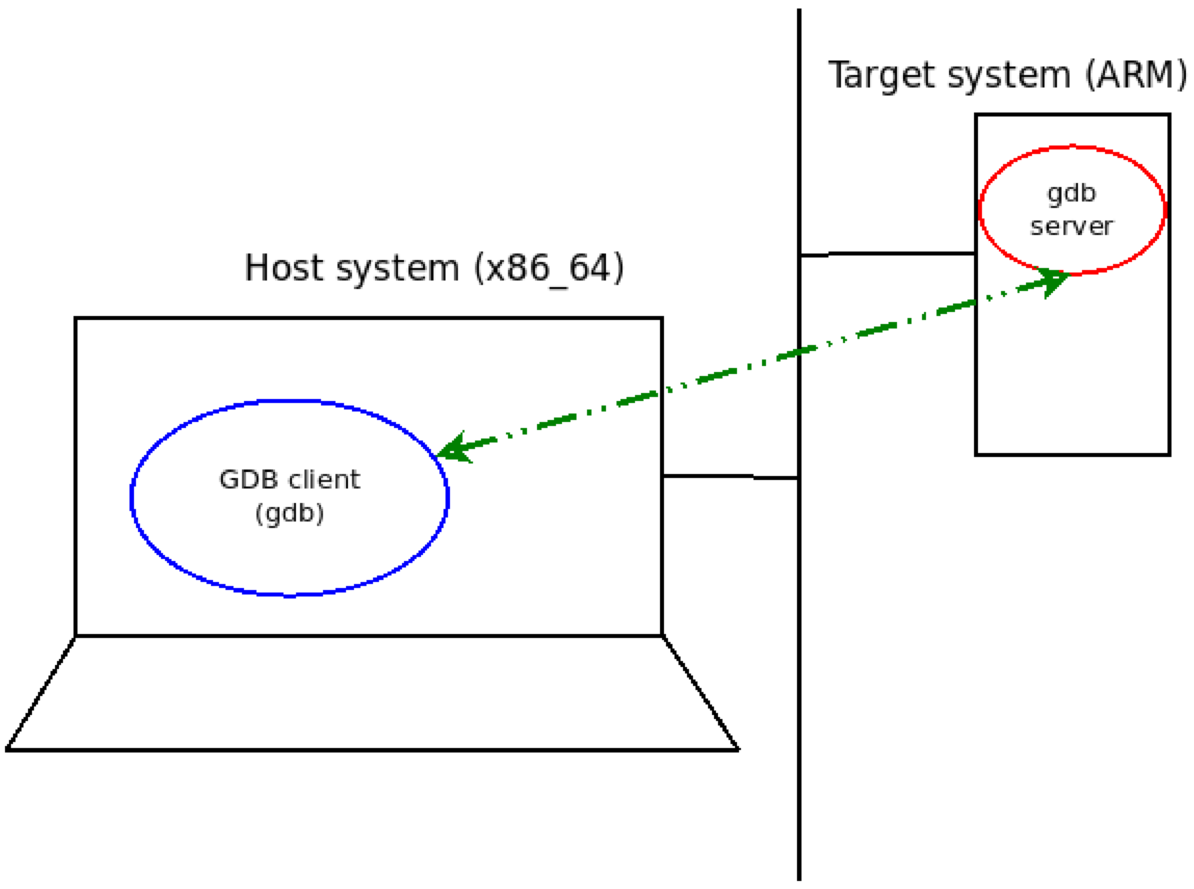


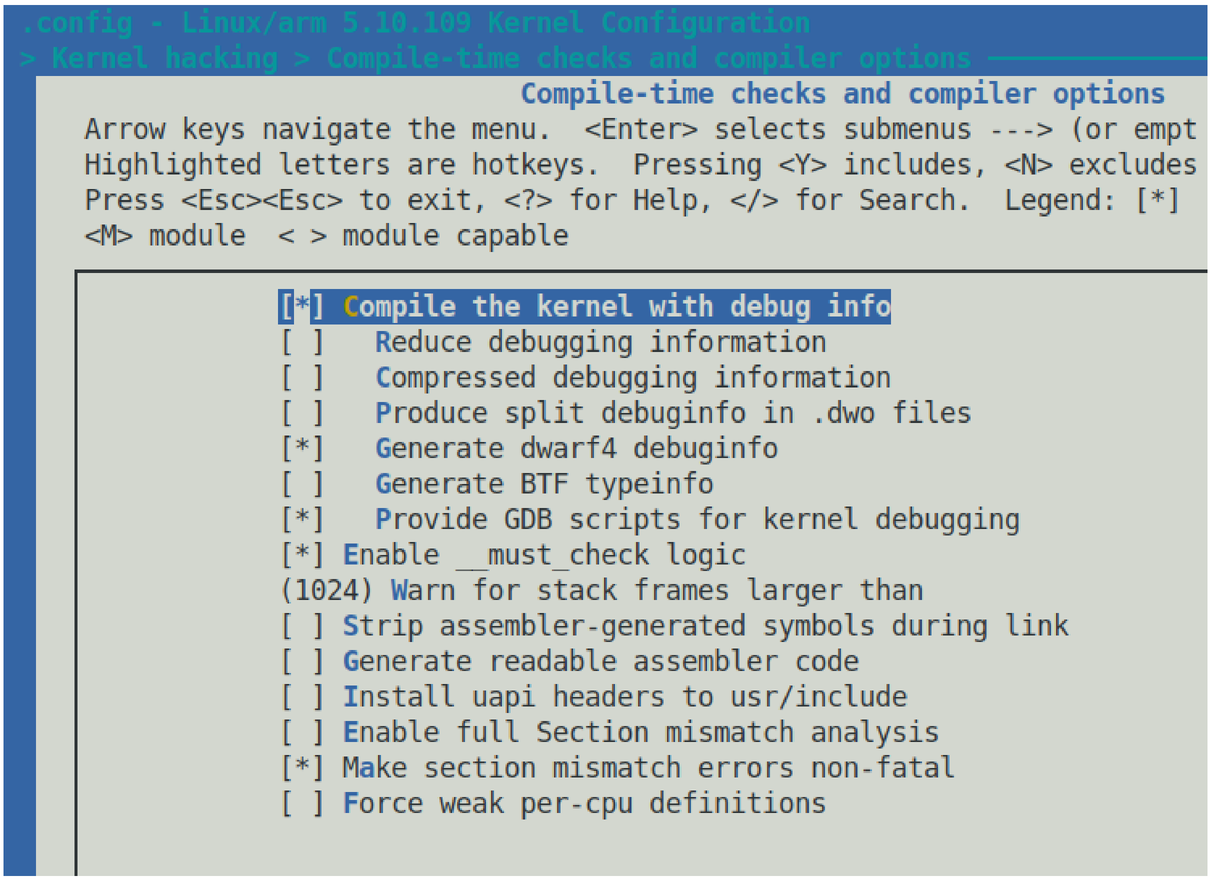
```
    while(!kthread_should_stop()) {
        //-----------------------------------
        pr_info("DELIBERATELY spinning on CPU core now...\n");

        if (likely(lockup_type == DO_SOFT_LOCKUP))
            spin_lock(&spinlock);
        else
            spin_lock_irq(&spinlock);

        while (i < 10000000000) { // adjust these arbit #s for your system if reqd..
            i ++;
            if (!(i%50000000))
                PRINT_CTX();
        }

        if (likely(lockup_type == DO_SOFT_LOCKUP))
            spin_unlock(&spinlock);
        else
            spin_unlock_irq(&spinlock);
        //-----------------------------------

        pr_info("FYI, I, kernel thread PID %d, am going to sleep now...\n",
            current->pid);
        set_current_state(TASK_INTERRUPTIBLE);
        schedule(); // yield the processor, go to sleep...
Message from syslogd@dbg-LKD at Mar 25 19:56:09 ...y due to either the
 kernel:[ 1528.659809] watchdog: BUG: soft lockup - CPU#2 stuck for 22s! [lkd/kt_stuck:3530]
         */
```



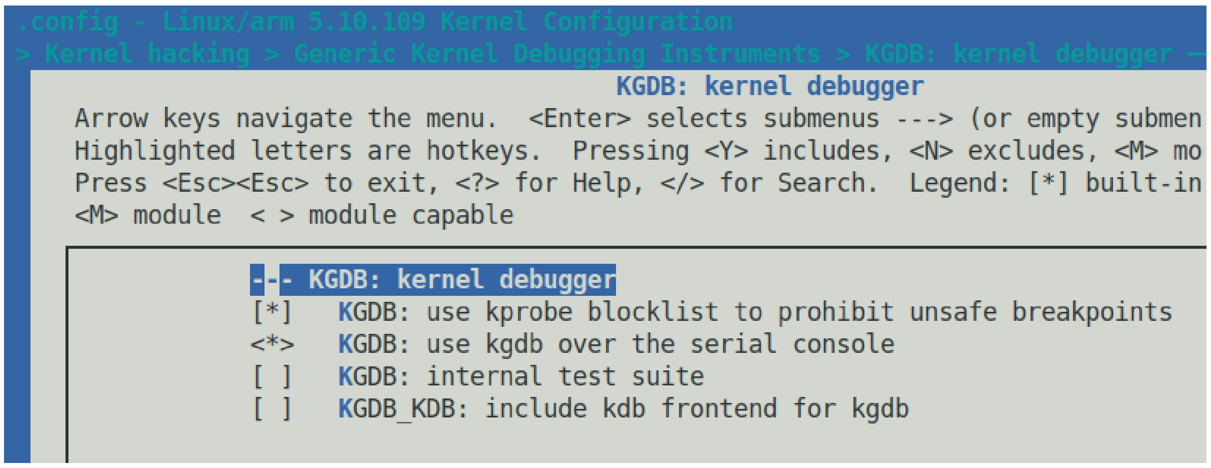
```
{
    struct st_ctx *priv = container_of(work, struct st_ctx, work);
    u64 i = 0;

    t2 = ktime_get_real_ns();
    pr_info("In our workq function: data=%d\n", priv->data);
    PRINT_CTX();
    SHOW_DELTA(t2, t1);

    /* Deliberately spin for a loooong while... causing the kernel softlockup
     * detector to swing into action!
     */
     pr_info("Deliberately locking up the cpu now!\n");
     //mdelay(1000*30);
     while (1)
        i += 3;
}

Message from syslogd@dbg-LKD at Mar 24 18:49:39 ...
 kernel:[29612.080043] BUG: workqueue lockup - pool cpus=2 node=0 flags=0x0 nice=0 stuck for
 166s!  ctx.data = INITIAL_VALUE;

Message from syslogd@dbg-LKD at Mar 24 18:50:10 ...
 kernel:[29642.797043] BUG: workqueue lockup - pool cpus=2 node=0 flags=0x0 nice=0 stuck for
 197s!
    /* Initialize our kernel timer */
Message from syslogd@dbg-LKD at Mar 24 18:50:40 ...s(exp_ms);
 kernel:[29673.522018] BUG: workqueue lockup - pool cpus=2 node=0 flags=0x0 nice=0 stuck for
 228s!                                                                  95,0-1        73%

Message from syslogd@dbg-LKD at Mar 24 18:51:11 ...
 kernel:[29704.249864] BUG: workqueue lockup - pool cpus=2 node=0 flags=0x0 nice=0 stuck for
 258s!
```

Chapter 11: Using Kernel GDB (KGDB)





```
.config - Linux/arm 5.10.109 Kernel Configuration
> Kernel hacking > Compile-time checks and compiler options ─────────
                          Compile-time checks and compiler options
   Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empt
   Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes
   Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*]
   <M> module  < > module capable

                [*] Compile the kernel with debug info
                [ ]   Reduce debugging information
                [ ]   Compressed debugging information
                [ ]   Produce split debuginfo in .dwo files
                [*]   Generate dwarf4 debuginfo
                [ ]   Generate BTF typeinfo
                [*]   Provide GDB scripts for kernel debugging
                [*] Enable __must_check logic
                (1024) Warn for stack frames larger than
                [ ] Strip assembler-generated symbols during link
                [ ] Generate readable assembler code
                [ ] Install uapi headers to usr/include
                [ ] Enable full Section mismatch analysis
                [*] Make section mismatch errors non-fatal
                [ ] Force weak per-cpu definitions
```



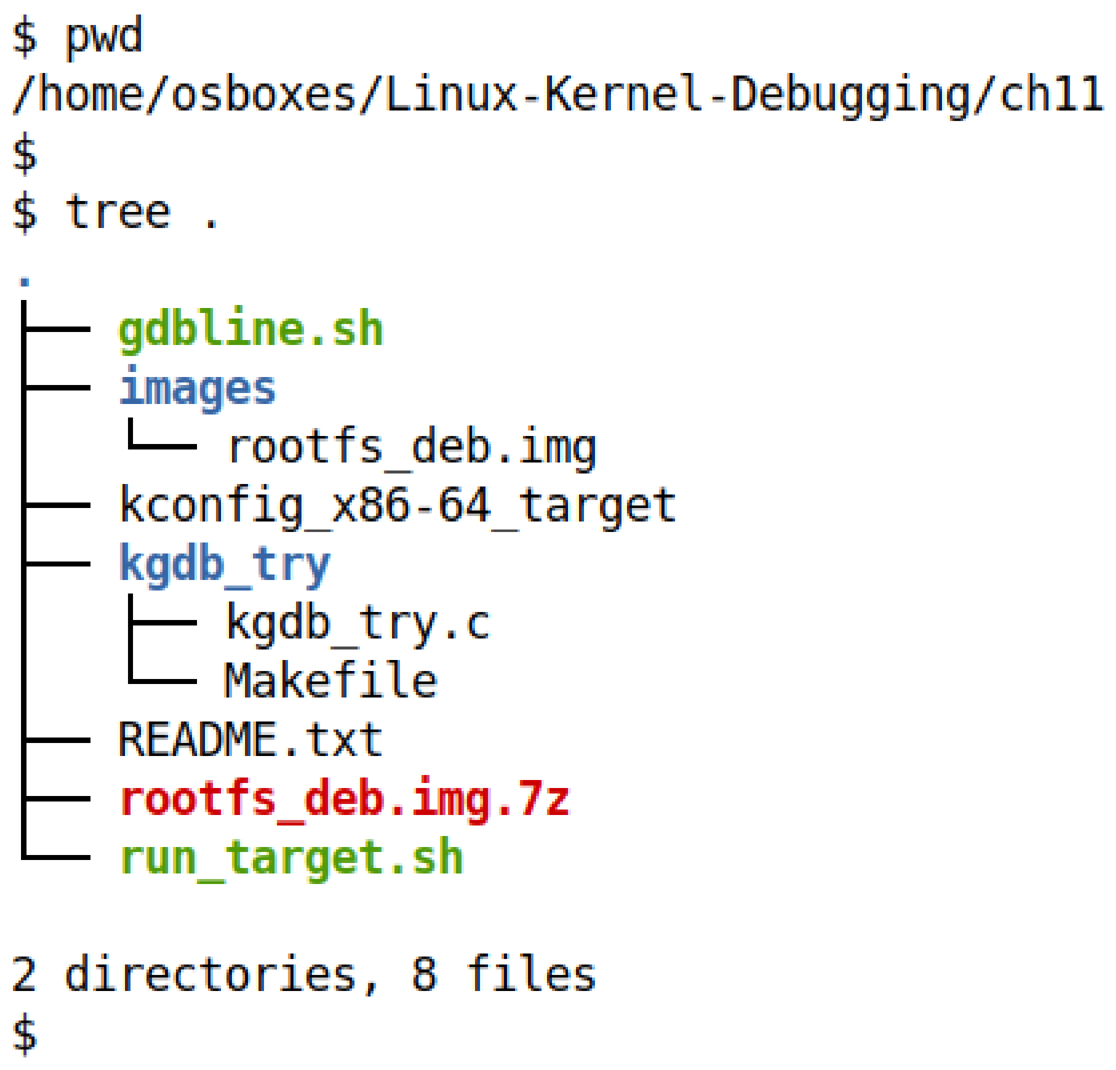
```
.config - Linux/arm 5.10.109 Kernel Configuration
> Kernel hacking > Generic Kernel Debugging Instruments > KGDB: kernel debugger ─
                                KGDB: kernel debugger
   Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submen
   Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> mo
   Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in
   <M> module  < > module capable

                --- KGDB: kernel debugger
                [*]   KGDB: use kprobe blocklist to prohibit unsafe breakpoints
                <*>   KGDB: use kgdb over the serial console
                [ ]   KGDB: internal test suite
                [ ]   KGDB_KDB: include kdb frontend for kgdb
```

```
$ qemu-system-arm -m 512 -M vexpress-a9 -smp 4,sockets=2 -kernel /home/letsdebug/seals_staging/seals_stagi
ng_vexpress/images/zImage -drive file=/home/letsdebug/seals_staging/seals_staging_vexpress/images/rfs.img,
if=sd,format=raw -append "console=ttyAMA0 rootfstype=ext4 root=/dev/mmcblk0 init=/sbin/init " -nographic -
no-reboot -audiodev id=none,driver=none -dtb /home/letsdebug/seals_staging/seals_staging_vexpress/images/v
express-v2p-ca9.dtb
audio: Device lm4549: audiodev default parameter is deprecated, please specify audiodev=none
Booting Linux on physical CPU 0x0
Linux version 5.10.109 (letsdebug@dbg-LKD) (arm-none-linux-gnueabihf-gcc (GNU Toolchain for the A-profile
Architecture 10.3-2021.07 (arm-10.29)) 10.3.1 20210621, GNU ld (GNU Toolchain for the A-profile Architectu
re 10.3-2021.07 (arm-10.29)) 2.36.1.20210621) #1 SMP Wed Apr 6 18:58:58 IST 2022
CPU: ARMv7 Processor [410fc090] revision 0 (ARMv7), cr=10c5387d
CPU: PIPT / VIPT nonaliasing data cache, VIPT nonaliasing instruction cache
OF: fdt: Machine model: V2P-CA9
Memory policy: Data cache writealloc
Reserved memory: created DMA memory pool at 0x4c000000, size 8 MiB
OF: reserved mem: initialized node vram@4c000000, compatible id shared-dma-pool
cma: Reserved 16 MiB at 0x7f000000
Zone ranges:
  Normal   [mem 0x0000000060000000-0x000000007fffffff]
Movable zone start for each node
Early memory node ranges
  node   0: [mem 0x0000000060000000-0x000000007fffffff]
```

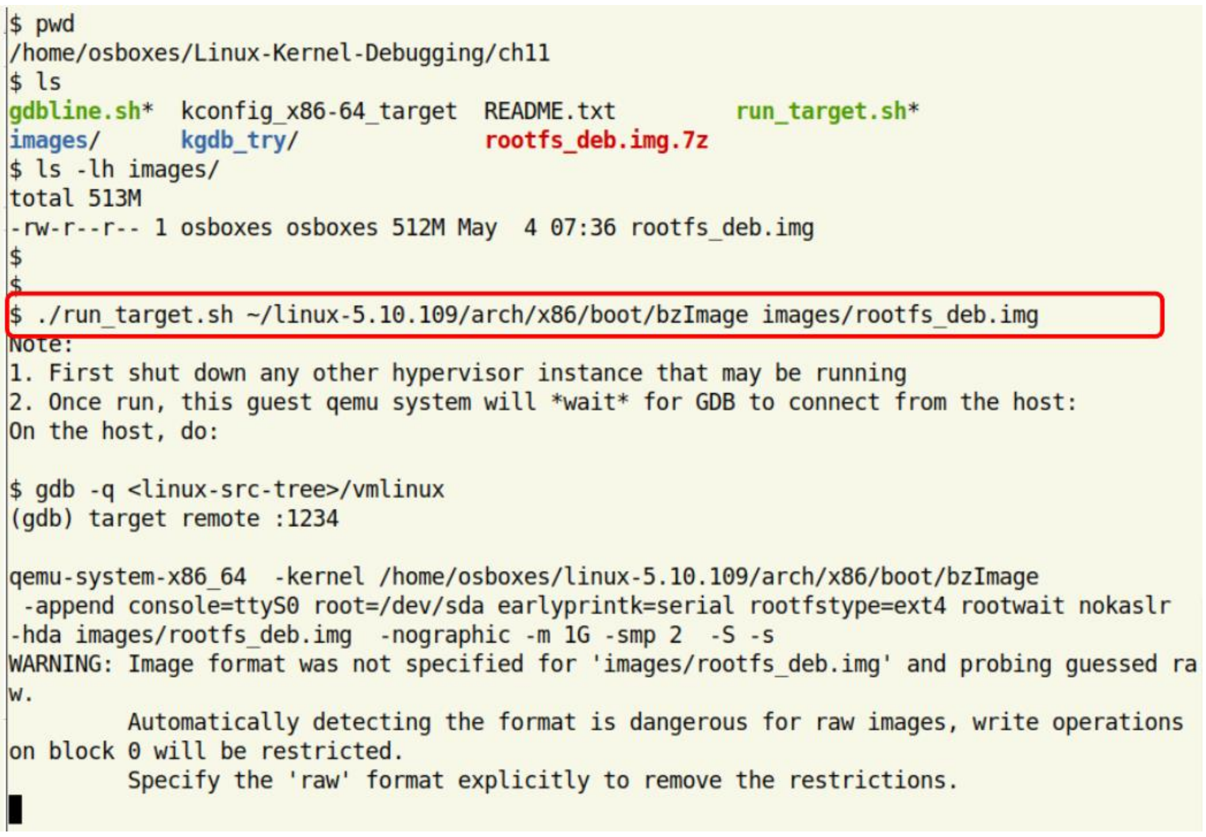
```
VFS: Mounted root (ext4 filesystem) readonly on device 179:0.
Freeing unused kernel memory: 1024K
Checked W+X mappings: passed, no W+X pages found
Run /sbin/init as init process
random: crng init done
SEALS: /etc/init.d/rcS running now ...
mount: mounting none on /sys/kernel/debug failed: No such file or directory
EXT4-fs (mmcblk0): re-mounted. Opts: (null)
Generic PHY 4e000000.ethernet-ffffffff:01: attached PHY driver [Generic PHY] (mii_bus:phy_addr=4e000000.et
hernet-ffffffff:01, irq=POLL)
smsc911x 4e000000.ethernet eth0: SMSC911x/921x identified at 0xa08b0000, IRQ: 30
/bin/sh: can't access tty; job control turned off
ARM / $
ARM / $ cat /proc/version
Linux version 5.10.109 (letsdebug@dbg-LKD) (arm-none-linux-gnueabihf-gcc (GNU Toolchain for the A-profile
Architecture 10.3-2021.07 (arm-10.29)) 10.3.1 20210621, GNU ld (GNU Toolchain for the A-profile Architectu
re 10.3-2021.07 (arm-10.29)) 2.36.1.20210621) #1 SMP Wed Apr 6 18:58:58 IST 2022
ARM / $
ARM / $ nproc
4
ARM / $ cat /proc/cpuinfo
processor       : 0
model name      : ARMv7 Processor rev 0 (v7l)
BogoMIPS        : 454.65
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpd32
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc09
CPU revision    : 0

processor       : 1
model name      : ARMv7 Processor rev 0 (v7l)
BogoMIPS        : 735.23
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpd32
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc09
CPU revision    : 0
```

```
(gdb) c
Continuing.

Breakpoint 2, register_netdev (dev=dev@entry=0x81014800) at net/core/dev.c:10238
10238           if (rtnl_lock_killable())
(gdb) bt
#0  register_netdev (dev=dev@entry=0x81014800) at net/core/dev.c:10238
#1  0x8065dc70 in smsc911x_drv_probe (pdev=0x81186410) at drivers/net/ethernet/smsc/smsc911x.c:2504
#2  0x805c4d88 in platform_drv_probe (_dev=0x81186410) at drivers/base/platform.c:761
#3  0x805c29c8 in really_probe (dev=dev@entry=0x81186410, drv=drv@entry=0x80c66498 <smsc911x_driver+20>) at drivers/base/dd.c:564
#4  0x805c306c in driver_probe_device (drv=drv@entry=0x80c66498 <smsc911x_driver+20>, dev=dev@entry=0x81186410)
    at drivers/base/dd.c:752
#5  0x805c3350 in device_driver_attach (drv=drv@entry=0x80c66498 <smsc911x_driver+20>, dev=dev@entry=0x81186410)
    at drivers/base/dd.c:1027
#6  0x805c33d8 in __driver_attach (data=0x80c66498 <smsc911x_driver+20>, dev=0x81186410) at drivers/base/dd.c:1104
#7  __driver_attach (dev=0x81186410, data=0x80c66498 <smsc911x_driver+20>) at drivers/base/dd.c:1058
#8  0x805c0a88 in bus_for_each_dev (bus=<optimized out>, start=start@entry=0x0, data=data@entry=0x80c66498 <smsc911x_driver+20>,
    fn=fn@entry=0x805c3358 <__driver_attach>) at drivers/base/bus.c:305
#9  0x805c2330 in driver_attach (drv=drv@entry=0x80c66498 <smsc911x_driver+20>) at drivers/base/dd.c:1120
#10 0x805c1de8 in bus_add_driver (drv=drv@entry=0x80c66498 <smsc911x_driver+20>) at drivers/base/bus.c:622
#11 0x805c3f04 in driver_register (drv=0x80c66498 <smsc911x_driver+20>) at drivers/base/driver.c:171
#12 0x80102064 in do_one_initcall (fn=0x80b23760 <smsc911x_init_module>) at init/main.c:1214
#13 0x80b012b8 in do_initcall_level (command_line=0x81118200 "console", level=6) at init/main.c:1287
#14 do_initcalls () at init/main.c:1303
#15 do_basic_setup () at init/main.c:1323
#16 kernel_init_freeable () at init/main.c:1525
#17 0x80862328 in kernel_init (unused=<optimized out>) at init/main.c:1412
#18 0x80100148 in ret_from_fork () at arch/arm/kernel/entry-common.S:155
Backtrace stopped: previous frame identical to this frame (corrupt stack?)
(gdb) l
10233    */
10234   int register_netdev(struct net_device *dev)
10235   {
10236           int err;
10237
10238           if (rtnl_lock_killable())
10239                   return -EINTR;
10240           err = register_netdevice(dev);
10241           rtnl_unlock();
10242           return err;
```

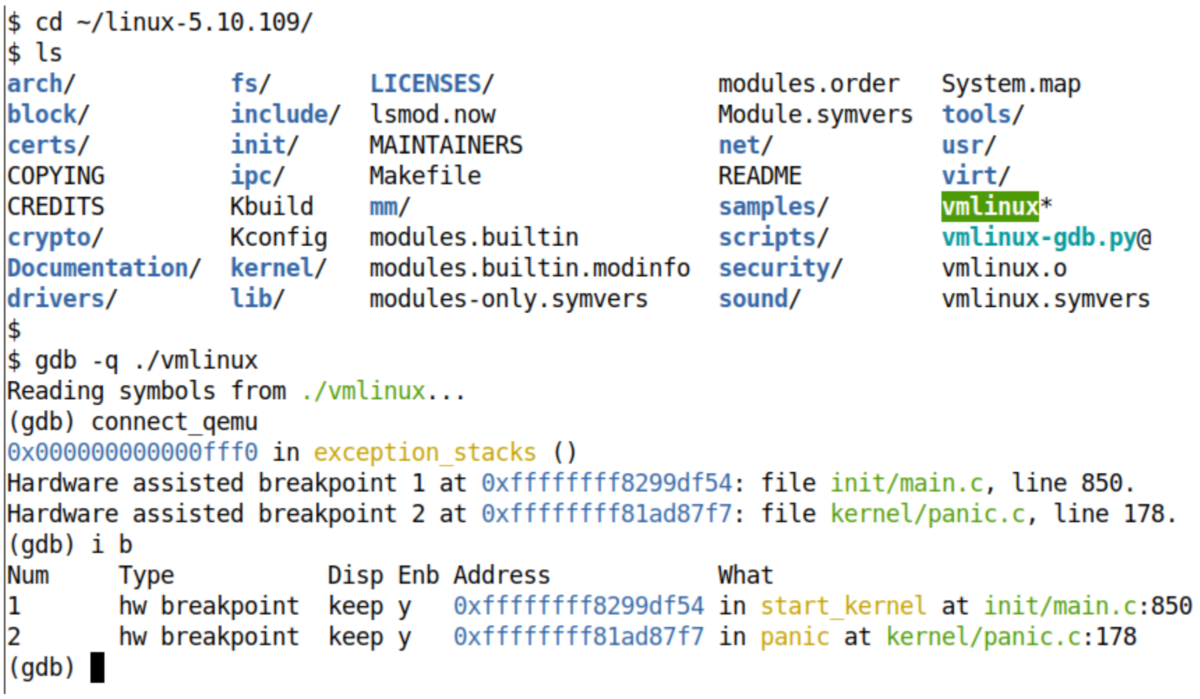
```
$ pwd
/home/osboxes/Linux-Kernel-Debugging/ch11
$ ls
gdbline.sh*  images/  kgdb_try/  rootfs_deb.img.7z  run_target.sh*
$ ls -l images/
total 524292
-rw-r--r-- 1 osboxes osboxes 536870912 May  3 17:20 rootfs_deb.img
$
$ sudo mount -o loop images/rootfs_deb.img /mnt/tmp
[sudo] password for osboxes:
$ ls /mnt/tmp/
bin/   dev/  home/  lib64/       media/  myprj/  proc/  run/   srv/  tmp/  var/
boot/  etc/  lib/   lost+found/  mnt/    opt/    root/  sbin/  sys/  usr/
$
$ ls /mnt/tmp/myprj/
doit*  gdbline.sh*  kgdb_try.ko
$
$ sudo umount /mnt/tmp
$
```



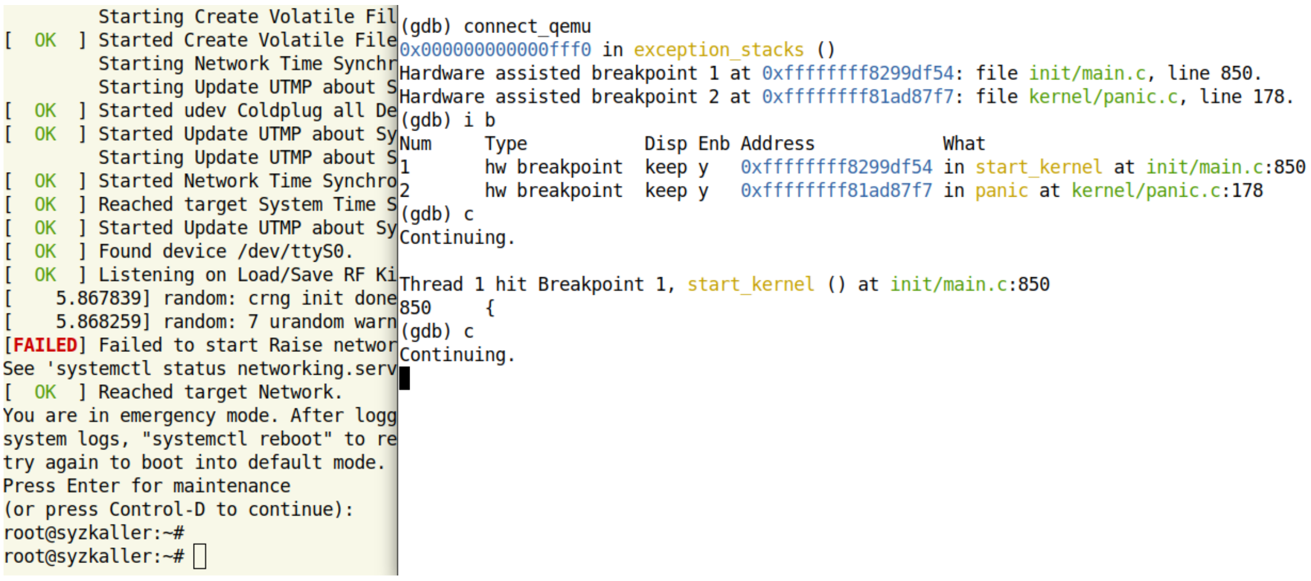
```
$ pwd
/home/osboxes/Linux-Kernel-Debugging/ch11
$
$ tree .
.
├── gdbline.sh
├── images
│   └── rootfs_deb.img
├── kconfig_x86-64_target
├── kgdb_try
│   ├── kgdb_try.c
│   └── Makefile
├── README.txt
├── rootfs_deb.img.7z
└── run_target.sh

2 directories, 8 files
$
```



```
$ pwd
/home/osboxes/Linux-Kernel-Debugging/ch11
$ ls
gdbline.sh*  kconfig_x86-64_target  README.txt        run_target.sh*
images/      kgdb_try/              rootfs_deb.img.7z
$ ls -lh images/
total 513M
-rw-r--r-- 1 osboxes osboxes 512M May  4 07:36 rootfs_deb.img
$
$
$ ./run_target.sh ~/linux-5.10.109/arch/x86/boot/bzImage images/rootfs_deb.img
Note:
1. First shut down any other hypervisor instance that may be running
2. Once run, this guest qemu system will *wait* for GDB to connect from the host:
On the host, do:

$ gdb -q <linux-src-tree>/vmlinux
(gdb) target remote :1234

qemu-system-x86_64  -kernel /home/osboxes/linux-5.10.109/arch/x86/boot/bzImage
 -append console=ttyS0 root=/dev/sda earlyprintk=serial rootfstype=ext4 rootwait nokaslr
-hda images/rootfs_deb.img  -nographic -m 1G -smp 2  -S -s
WARNING: Image format was not specified for 'images/rootfs_deb.img' and probing guessed ra
w.
         Automatically detecting the format is dangerous for raw images, write operations
on block 0 will be restricted.
         Specify the 'raw' format explicitly to remove the restrictions.
█
```

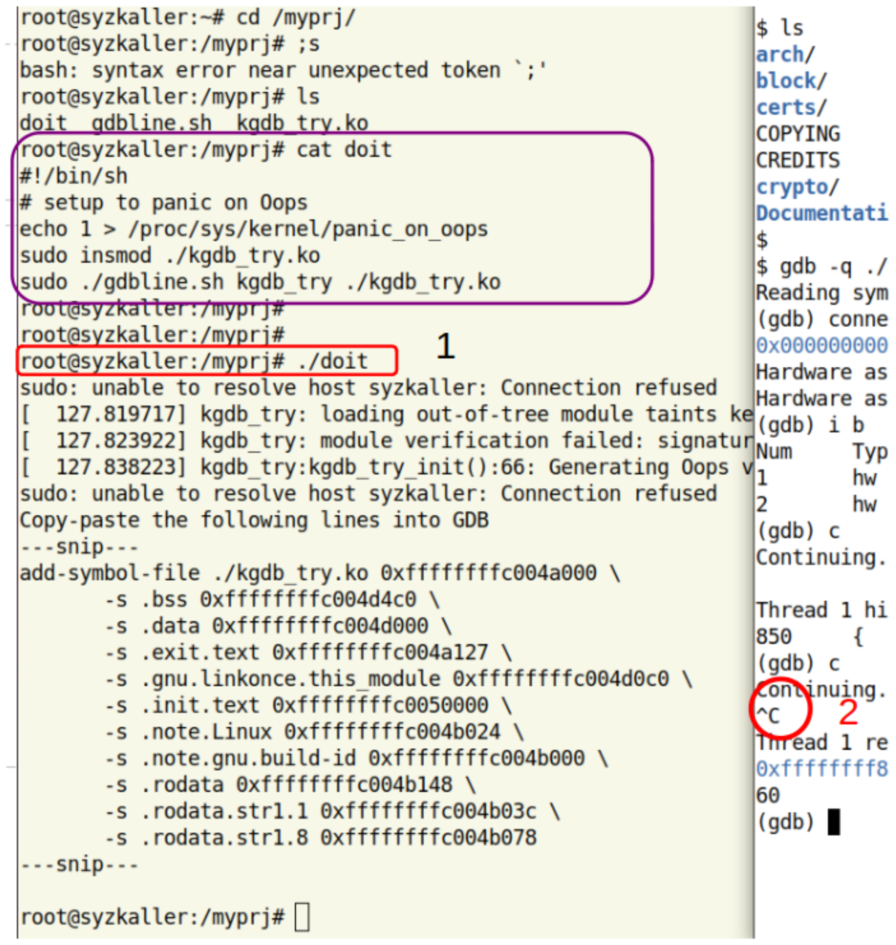


```
$ cd ~/linux-5.10.109/
$ ls
arch/           fs/        LICENSES/               modules.order    System.map
block/          include/   lsmod.now               Module.symvers   tools/
certs/          init/      MAINTAINERS             net/             usr/
COPYING         ipc/       Makefile                README           virt/
CREDITS         Kbuild     mm/                     samples/         vmlinux*
crypto/         Kconfig    modules.builtin         scripts/         vmlinux-gdb.py@
Documentation/  kernel/    modules.builtin.modinfo security/        vmlinux.o
drivers/        lib/       modules-only.symvers    sound/           vmlinux.symvers
$
$ gdb -q ./vmlinux
Reading symbols from ./vmlinux...
(gdb) connect_qemu
0x000000000000fff0 in exception_stacks ()
Hardware assisted breakpoint 1 at 0xffffffff8299df54: file init/main.c, line 850.
Hardware assisted breakpoint 2 at 0xffffffff81ad87f7: file kernel/panic.c, line 178.
(gdb) i b
Num     Type           Disp Enb Address            What
1       hw breakpoint  keep y   0xffffffff8299df54 in start_kernel at init/main.c:850
2       hw breakpoint  keep y   0xffffffff81ad87f7 in panic at kernel/panic.c:178
(gdb) █
```



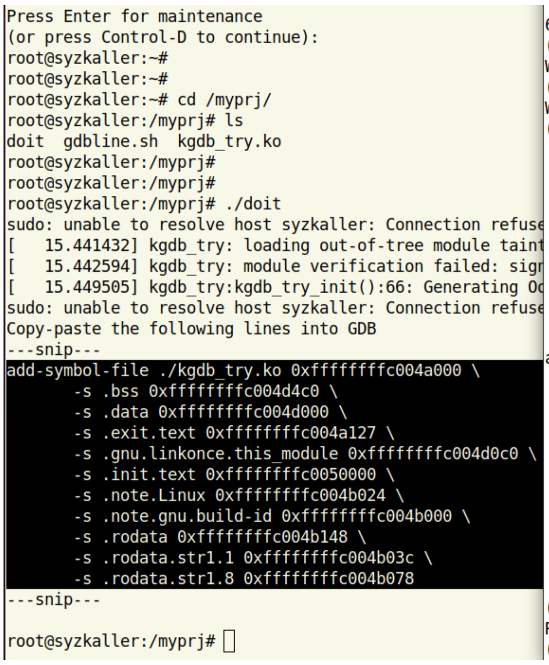
```
         Starting Create Volatile Fil
[  OK  ] Started Create Volatile File
         Starting Network Time Synchr
         Starting Update UTMP about S
[  OK  ] Started udev Coldplug all De
[  OK  ] Started Update UTMP about Sy
         Starting Update UTMP about S
[  OK  ] Started Network Time Synchro
[  OK  ] Reached target System Time S
[  OK  ] Started Update UTMP about Sy
[  OK  ] Found device /dev/ttyS0.
[  OK  ] Listening on Load/Save RF Ki
[    5.867839] random: crng init done
[    5.868259] random: 7 urandom warn
[FAILED] Failed to start Raise networ
See 'systemctl status networking.serv
[  OK  ] Reached target Network.
You are in emergency mode. After logg
system logs, "systemctl reboot" to re
try again to boot into default mode.
Press Enter for maintenance
(or press Control-D to continue):
root@syzkaller:~#
root@syzkaller:~#
```

```
(gdb) connect_qemu
0x000000000000fff0 in exception_stacks ()
Hardware assisted breakpoint 1 at 0xffffffff8299df54: file init/main.c, line 850.
Hardware assisted breakpoint 2 at 0xffffffff81ad87f7: file kernel/panic.c, line 178.
(gdb) i b
Num     Type           Disp Enb Address            What
1       hw breakpoint  keep y   0xffffffff8299df54 in start_kernel at init/main.c:850
2       hw breakpoint  keep y   0xffffffff81ad87f7 in panic at kernel/panic.c:178
(gdb) c
Continuing.

Thread 1 hit Breakpoint 1, start_kernel () at init/main.c:850
850     {
(gdb) c
Continuing.
```

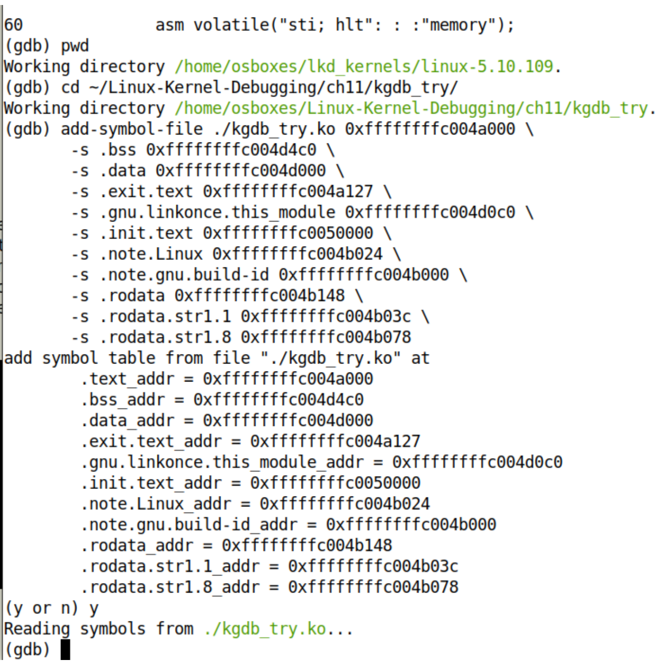


```
root@syzkaller:~# cd /myprj/
root@syzkaller:/myprj# ;s
bash: syntax error near unexpected token `;'
root@syzkaller:/myprj# ls
doit  gdbline.sh  kgdb_try.ko
root@syzkaller:/myprj# cat doit
#!/bin/sh
# setup to panic on Oops
echo 1 > /proc/sys/kernel/panic_on_oops
sudo insmod ./kgdb_try.ko
sudo ./gdbline.sh kgdb_try ./kgdb_try.ko
root@syzkaller:/myprj#
root@syzkaller:/myprj#
root@syzkaller:/myprj# ./doit
sudo: unable to resolve host syzkaller: Connection refused
[  127.819717] kgdb_try: loading out-of-tree module taints ke
[  127.823922] kgdb_try: module verification failed: signatur
[  127.838223] kgdb_try:kgdb_try_init():66: Generating Oops v
sudo: unable to resolve host syzkaller: Connection refused
Copy-paste the following lines into GDB
---snip---
add-symbol-file ./kgdb_try.ko 0xffffffffc004a000 \
       -s .bss 0xffffffffc004d4c0 \
       -s .data 0xffffffffc004d000 \
       -s .exit.text 0xffffffffc004a127 \
       -s .gnu.linkonce.this_module 0xffffffffc004d0c0 \
       -s .init.text 0xffffffffc0050000 \
       -s .note.Linux 0xffffffffc004b024 \
       -s .note.gnu.build-id 0xffffffffc004b000 \
       -s .rodata 0xffffffffc004b148 \
       -s .rodata.str1.1 0xffffffffc004b03c \
       -s .rodata.str1.8 0xffffffffc004b078
---snip---

root@syzkaller:/myprj#
```

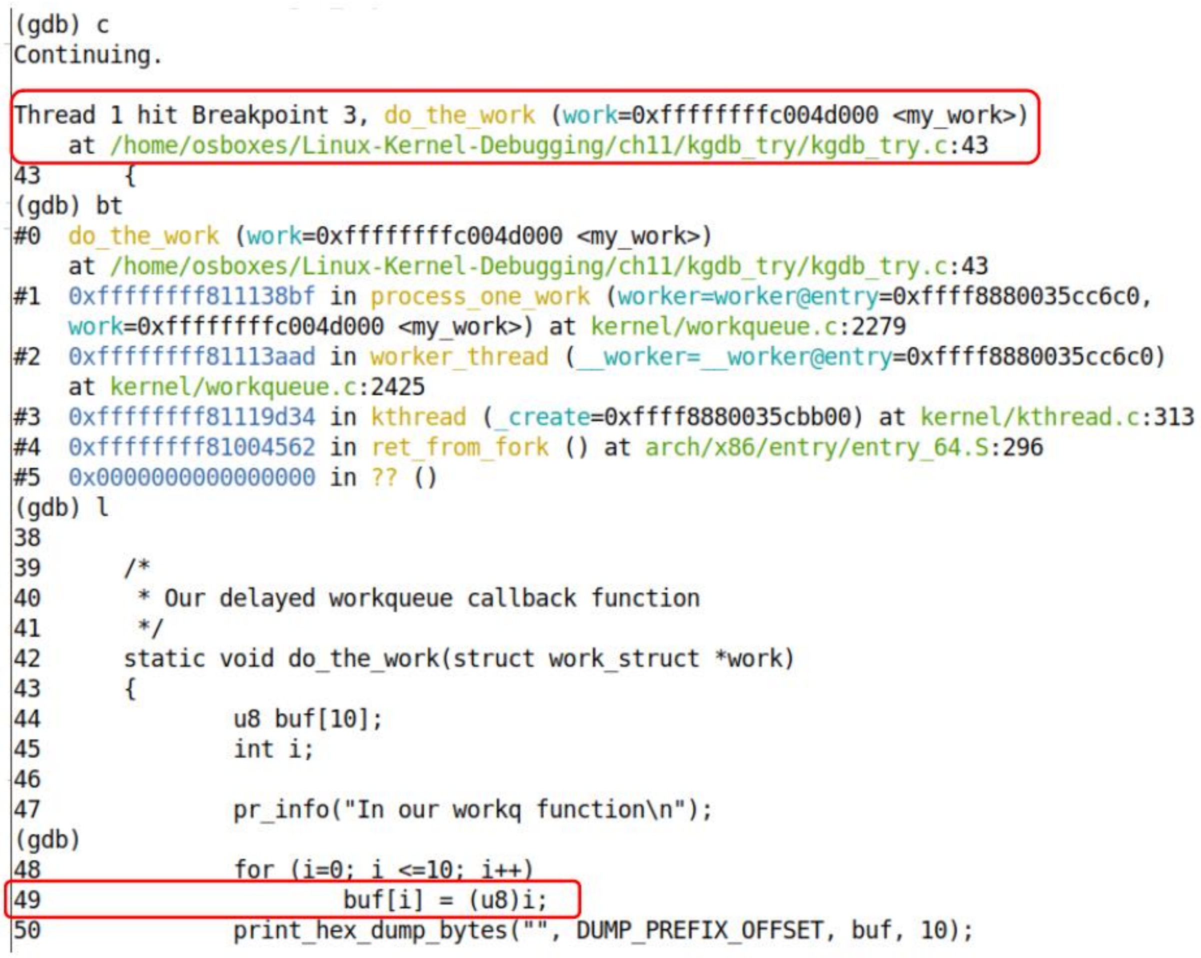
```
$ ls
arch/
block/
certs/
COPYING
CREDITS
crypto/
Documentati
$
$ gdb -q ./
Reading sym
(gdb) conne
0x000000000
Hardware as
Hardware as
(gdb) i b
Num     Typ
1       hw
2       hw
(gdb) c
Continuing.

Thread 1 hi
850     {
(gdb) c
Continuing.
^C
Thread 1 re
0xffffffff8
60
(gdb)
```



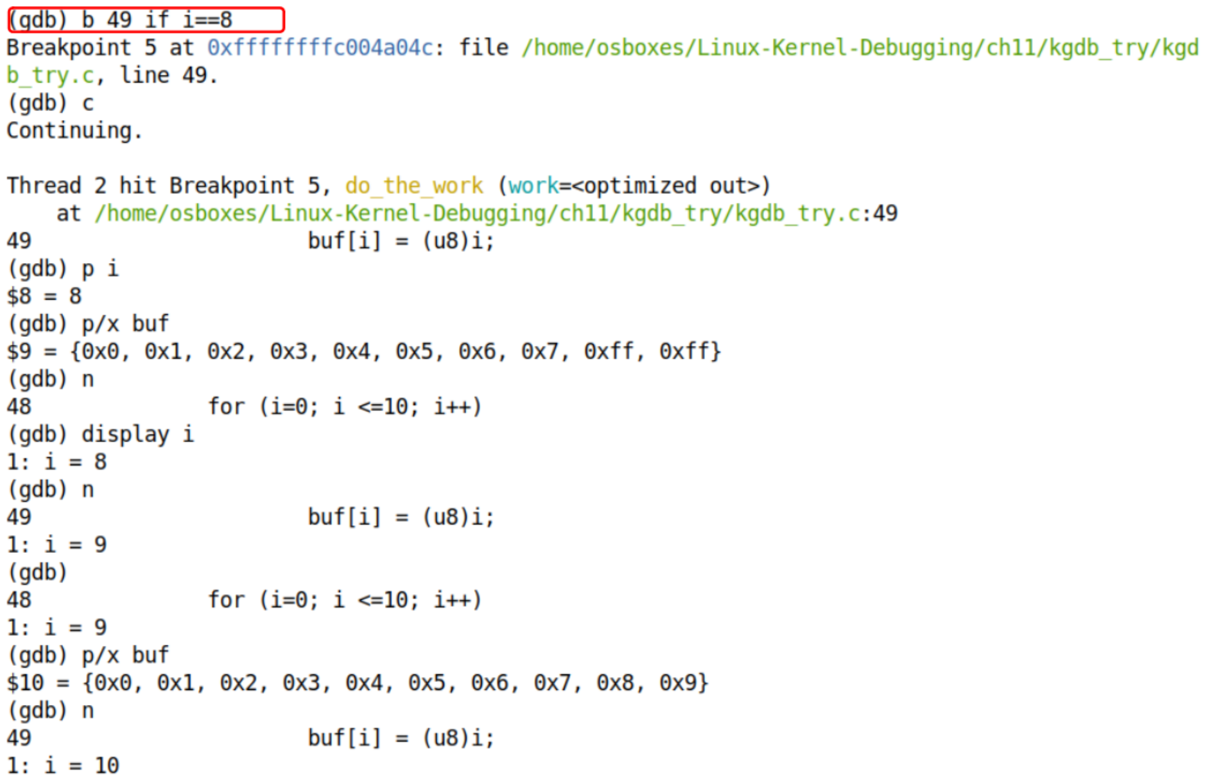
```
Press Enter for maintenance
(or press Control-D to continue):
root@syzkaller:~#
root@syzkaller:~#
root@syzkaller:~# cd /myprj/
root@syzkaller:/myprj# ls
doit  gdbline.sh  kgdb_try.ko
root@syzkaller:/myprj#
root@syzkaller:/myprj#
root@syzkaller:/myprj# ./doit
sudo: unable to resolve host syzkaller: Connection refuse
[   15.441432] kgdb_try: loading out-of-tree module taint
[   15.442594] kgdb_try: module verification failed: sigr
[   15.449505] kgdb_try:kgdb_try_init():66: Generating Oo
sudo: unable to resolve host syzkaller: Connection refuse
Copy-paste the following lines into GDB
---snip---
add-symbol-file ./kgdb_try.ko 0xffffffffc004a000 \
       -s .bss 0xffffffffc004d4c0 \
       -s .data 0xffffffffc004d000 \
       -s .exit.text 0xffffffffc004a127 \
       -s .gnu.linkonce.this_module 0xffffffffc004d0c0 \
       -s .init.text 0xffffffffc0050000 \
       -s .note.Linux 0xffffffffc004b024 \
       -s .note.gnu.build-id 0xffffffffc004b000 \
       -s .rodata 0xffffffffc004b148 \
       -s .rodata.str1.1 0xffffffffc004b03c \
       -s .rodata.str1.8 0xffffffffc004b078
---snip---

root@syzkaller:/myprj# 
```



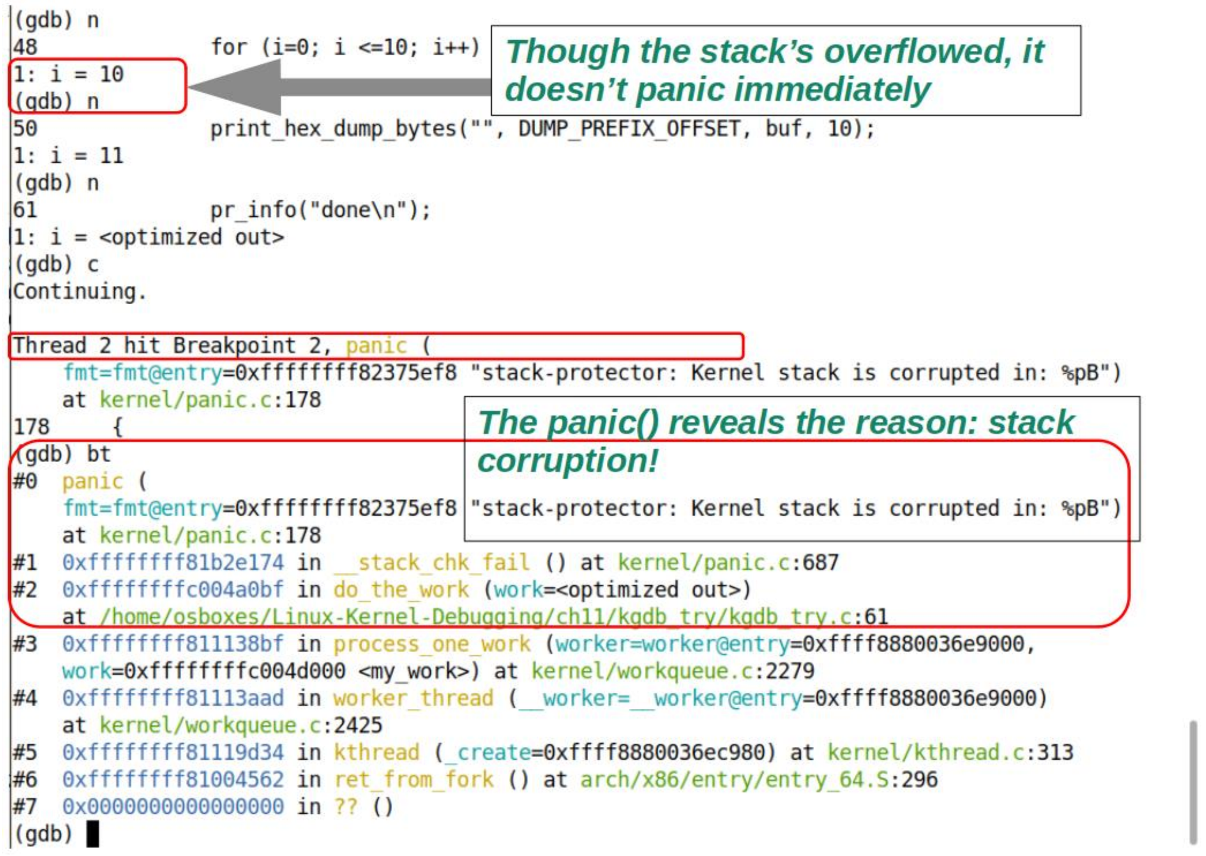
```
60          asm volatile("sti; hlt": : :"memory");
(gdb) pwd
Working directory /home/osboxes/lkd_kernels/linux-5.10.109.
(gdb) cd ~/Linux-Kernel-Debugging/ch11/kgdb_try/
Working directory /home/osboxes/Linux-Kernel-Debugging/ch11/kgdb_try.
(gdb) add-symbol-file ./kgdb_try.ko 0xffffffffc004a000 \
       -s .bss 0xffffffffc004d4c0 \
       -s .data 0xffffffffc004d000 \
       -s .exit.text 0xffffffffc004a127 \
       -s .gnu.linkonce.this_module 0xffffffffc004d0c0 \
       -s .init.text 0xffffffffc0050000 \
       -s .note.Linux 0xffffffffc004b024 \
       -s .note.gnu.build-id 0xffffffffc004b000 \
       -s .rodata 0xffffffffc004b148 \
       -s .rodata.str1.1 0xffffffffc004b03c \
       -s .rodata.str1.8 0xffffffffc004b078
add symbol table from file "./kgdb_try.ko" at
        .text_addr = 0xffffffffc004a000
        .bss_addr = 0xffffffffc004d4c0
        .data_addr = 0xffffffffc004d000
        .exit.text_addr = 0xffffffffc004a127
        .gnu.linkonce.this_module_addr = 0xffffffffc004d0c0
        .init.text_addr = 0xffffffffc0050000
        .note.Linux_addr = 0xffffffffc004b024
        .note.gnu.build-id_addr = 0xffffffffc004b000
        .rodata_addr = 0xffffffffc004b148
        .rodata.str1.1_addr = 0xffffffffc004b03c
        .rodata.str1.8_addr = 0xffffffffc004b078
(y or n) y
Reading symbols from ./kgdb_try.ko...
(gdb) 
```



```
(gdb) c
Continuing.

Thread 1 hit Breakpoint 3, do_the_work (work=0xffffffffc004d000 <my_work>)
    at /home/osboxes/Linux-Kernel-Debugging/ch11/kgdb_try/kgdb_try.c:43
43          {
(gdb) bt
#0  do_the_work (work=0xffffffffc004d000 <my_work>)
    at /home/osboxes/Linux-Kernel-Debugging/ch11/kgdb_try/kgdb_try.c:43
#1  0xffffffff811138bf in process_one_work (worker=worker@entry=0xffff8880035cc6c0,
    work=0xffffffffc004d000 <my_work>) at kernel/workqueue.c:2279
#2  0xffffffff81113aad in worker_thread (__worker=__worker@entry=0xffff8880035cc6c0)
    at kernel/workqueue.c:2425
#3  0xffffffff81119d34 in kthread (_create=0xffff8880035cbb00) at kernel/kthread.c:313
#4  0xffffffff81004562 in ret_from_fork () at arch/x86/entry/entry_64.S:296
#5  0x0000000000000000 in ?? ()
(gdb) l
38
39          /*
40           * Our delayed workqueue callback function
41           */
42          static void do_the_work(struct work_struct *work)
43          {
44                  u8 buf[10];
45                  int i;
46
47                  pr_info("In our workq function\n");
(gdb)
48                  for (i=0; i <=10; i++)
49                          buf[i] = (u8)i;
50                  print_hex_dump_bytes("", DUMP_PREFIX_OFFSET, buf, 10);
```



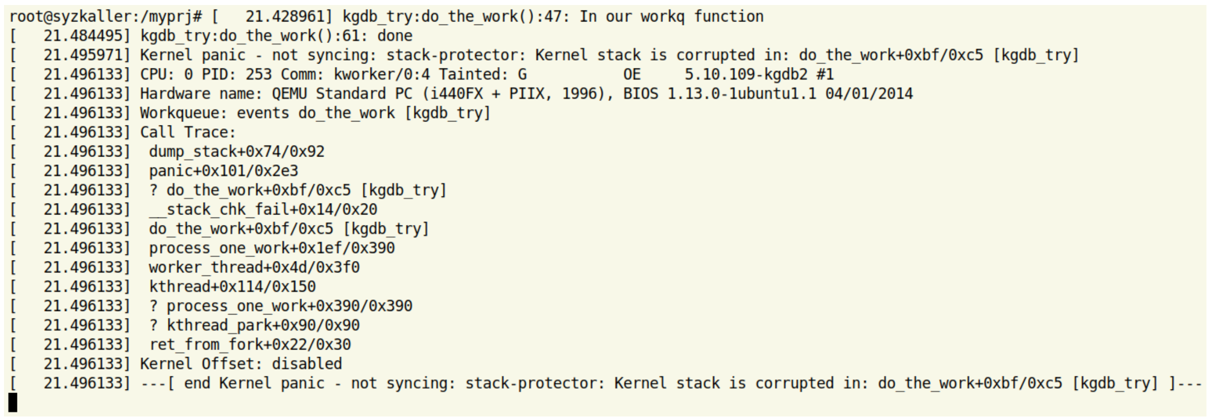
```
(gdb) b 49 if i==8
Breakpoint 5 at 0xffffffffc004a04c: file /home/osboxes/Linux-Kernel-Debugging/ch11/kgdb_try/kgd
b_try.c, line 49.
(gdb) c
Continuing.

Thread 2 hit Breakpoint 5, do_the_work (work=<optimized out>)
    at /home/osboxes/Linux-Kernel-Debugging/ch11/kgdb_try/kgdb_try.c:49
49                      buf[i] = (u8)i;
(gdb) p i
$8 = 8
(gdb) p/x buf
$9 = {0x0, 0x1, 0x2, 0x3, 0x4, 0x5, 0x6, 0x7, 0xff, 0xff}
(gdb) n
48              for (i=0; i <=10; i++)
(gdb) display i
1: i = 8
(gdb) n
49                      buf[i] = (u8)i;
1: i = 9
(gdb)
48              for (i=0; i <=10; i++)
1: i = 9
(gdb) p/x buf
$10 = {0x0, 0x1, 0x2, 0x3, 0x4, 0x5, 0x6, 0x7, 0x8, 0x9}
(gdb) n
49                      buf[i] = (u8)i;
1: i = 10
```



```
(gdb) n
48              for (i=0; i <=10; i++)
1: i = 10
(gdb) n
50              print_hex_dump_bytes("", DUMP_PREFIX_OFFSET, buf, 10);
1: i = 11
(gdb) n
61              pr_info("done\n");
1: i = <optimized out>
(gdb) c
Continuing.

Thread 2 hit Breakpoint 2, panic (
    fmt=fmt@entry=0xffffffff82375ef8 "stack-protector: Kernel stack is corrupted in: %pB")
    at kernel/panic.c:178
178     {
(gdb) bt
#0  panic (
    fmt=fmt@entry=0xffffffff82375ef8 "stack-protector: Kernel stack is corrupted in: %pB")
    at kernel/panic.c:178
#1  0xffffffff81b2e174 in __stack_chk_fail () at kernel/panic.c:687
#2  0xffffffffc004a0bf in do_the_work (work=<optimized out>)
    at /home/osboxes/Linux-Kernel-Debugging/ch11/kgdb_try/kgdb_try.c:61
#3  0xffffffff811138bf in process_one_work (worker=worker@entry=0xffff8880036e9000,
    work=0xffffffffc004d000 <my_work>) at kernel/workqueue.c:2279
#4  0xffffffff81113aad in worker_thread (__worker=__worker@entry=0xffff8880036e9000)
    at kernel/workqueue.c:2425
#5  0xffffffff81119d34 in kthread (_create=0xffff8880036ec980) at kernel/kthread.c:313
#6  0xffffffff81004562 in ret_from_fork () at arch/x86/entry/entry_64.S:296
#7  0x0000000000000000 in ?? ()
(gdb)
```

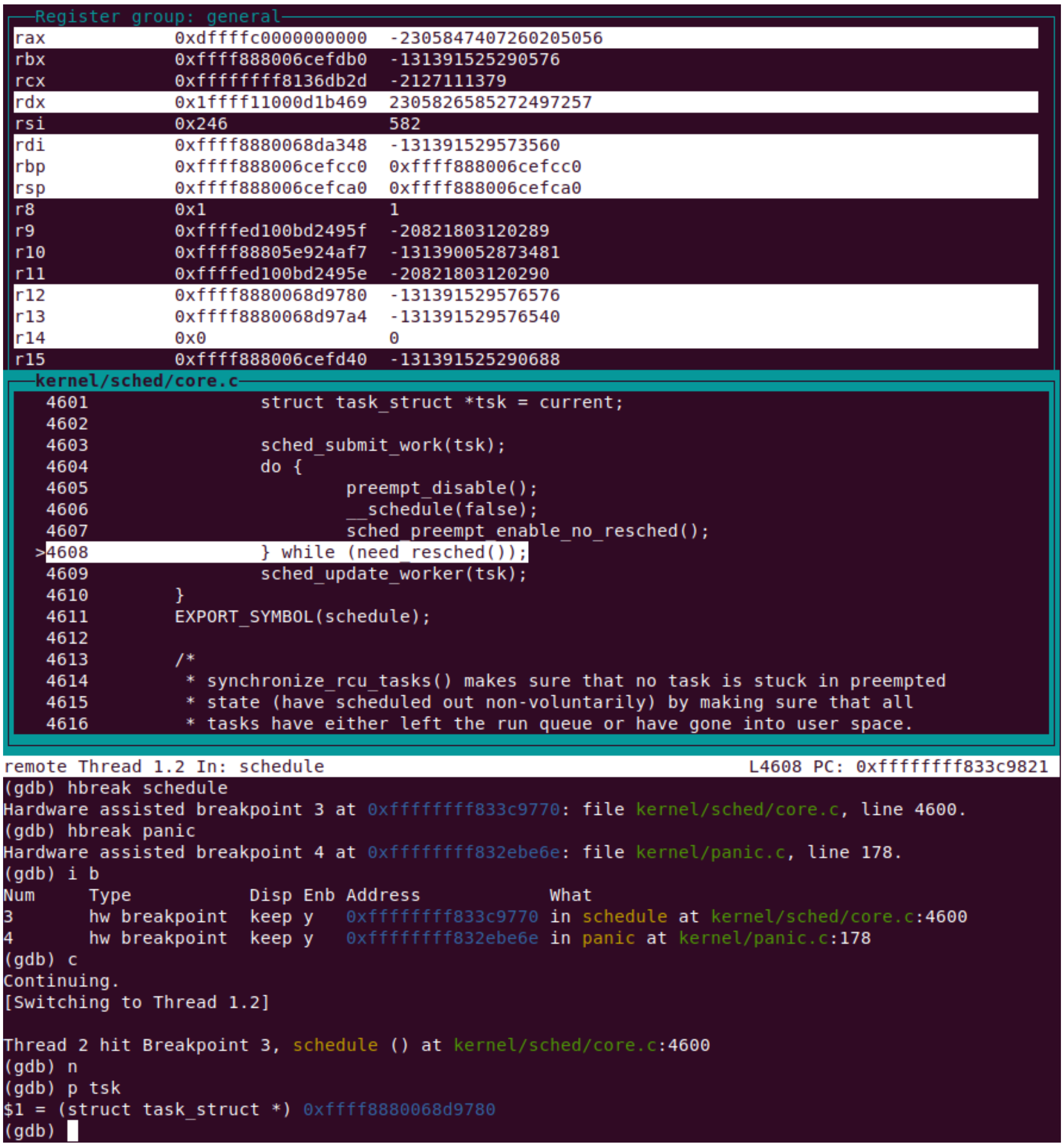
Though the stack's overflowed, it doesn't panic immediately

The panic() reveals the reason: stack corruption!



```
root@syzkaller:/myprj# [   21.428961] kgdb_try:do_the_work():47: In our workq function
[   21.484495] kgdb_try:do_the_work():61: done
[   21.495971] Kernel panic - not syncing: stack-protector: Kernel stack is corrupted in: do_the_work+0xbf/0xc5 [kgdb_try]
[   21.496133] CPU: 0 PID: 253 Comm: kworker/0:4 Tainted: G           OE     5.10.109-kgdb2 #1
[   21.496133] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.13.0-1ubuntu1.1 04/01/2014
[   21.496133] Workqueue: events do_the_work [kgdb_try]
[   21.496133] Call Trace:
[   21.496133]  dump_stack+0x74/0x92
[   21.496133]  panic+0x101/0x2e3
[   21.496133]  ? do_the_work+0xbf/0xc5 [kgdb_try]
[   21.496133]  __stack_chk_fail+0x14/0x20
[   21.496133]  do_the_work+0xbf/0xc5 [kgdb_try]
[   21.496133]  process_one_work+0x1ef/0x390
[   21.496133]  worker_thread+0x4d/0x3f0
[   21.496133]  kthread+0x114/0x150
[   21.496133]  ? process_one_work+0x390/0x390
[   21.496133]  ? kthread_park+0x90/0x90
[   21.496133]  ret_from_fork+0x22/0x30
[   21.496133] Kernel Offset: disabled
[   21.496133] ---[ end Kernel panic - not syncing: stack-protector: Kernel stack is corrupted in: do_the_work+0xbf/0xc5 [kgdb_try] ]---
```

```
(gdb) apropos lx
function lx_clk_core_lookup -- Find struct clk_core by name
function lx_current -- Return current task.
function lx_device_find_by_bus_name -- Find struct device by bus and name (both strings)
function lx_device_find_by_class_name -- Find struct device by class and name (both strings)
function lx_module -- Find module by name and return the module variable.
function lx_per_cpu -- Return per-cpu variable.
function lx_rb_first -- Lookup and return a node from an RBTree
function lx_rb_last -- Lookup and return a node from an RBTree.
function lx_rb_next -- Lookup and return a node from an RBTree.
function lx_rb_prev -- Lookup and return a node from an RBTree.
function lx_task_by_pid -- Find Linux task by PID and return the task_struct variable.
function lx_thread_info -- Calculate Linux thread_info from task variable.
function lx_thread_info_by_pid -- Calculate Linux thread_info from task variable found by pid
lx-clk-summary -- Print clk tree summary
lx-cmdline --  Report the Linux Commandline used in the current kernel.
lx-configdump -- Output kernel config to the filename specified as the command
lx-cpus -- List CPU status arrays
lx-device-list-bus -- Print devices on a bus (or all buses if not specified)
lx-device-list-class -- Print devices in a class (or all classes if not specified)
lx-device-list-tree -- Print a device and its children recursively
lx-dmesg -- Print Linux kernel log buffer.
lx-fdtdump -- Output Flattened Device Tree header and dump FDT blob to the filename
lx-genpd-summary -- Print genpd summary
lx-iomem -- Identify the IO memory resource locations defined by the kernel
lx-ioports -- Identify the IO port resource locations defined by the kernel
lx-list-check -- Verify a list consistency
lx-lsmod -- List currently loaded modules.
lx-mounts -- Report the VFS mounts of the current process namespace.
lx-ps -- Dump Linux tasks.
lx-symbols -- (Re-)load symbols of Linux kernel and currently loaded modules.
lx-timerlist -- Print /proc/timer_list
lx-version --  Report the Linux Version of the current kernel.
```



```
Register group: general
rax            0xdffffc0000000000  -2305847407260205056
rbx            0xffff888006cefdb0  -131391525290576
rcx            0xffffffff8136db2d  -2127111379
rdx            0x1ffff11000d1b469  2305826585272497257
rsi            0x246               582
rdi            0xffff8880068da348  -131391529573560
rbp            0xffff888006cefcc0  0xffff888006cefcc0
rsp            0xffff888006cefca0  0xffff888006cefca0
r8             0x1                 1
r9             0xffffed100bd2495f  -20821803120289
r10            0xffff88805e924af7  -131390052873481
r11            0xffffed100bd2495e  -20821803120290
r12            0xffff8880068d9780  -131391529576576
r13            0xffff8880068d97a4  -131391529576540
r14            0x0                 0
r15            0xffff888006cefd40  -131391525290688
```

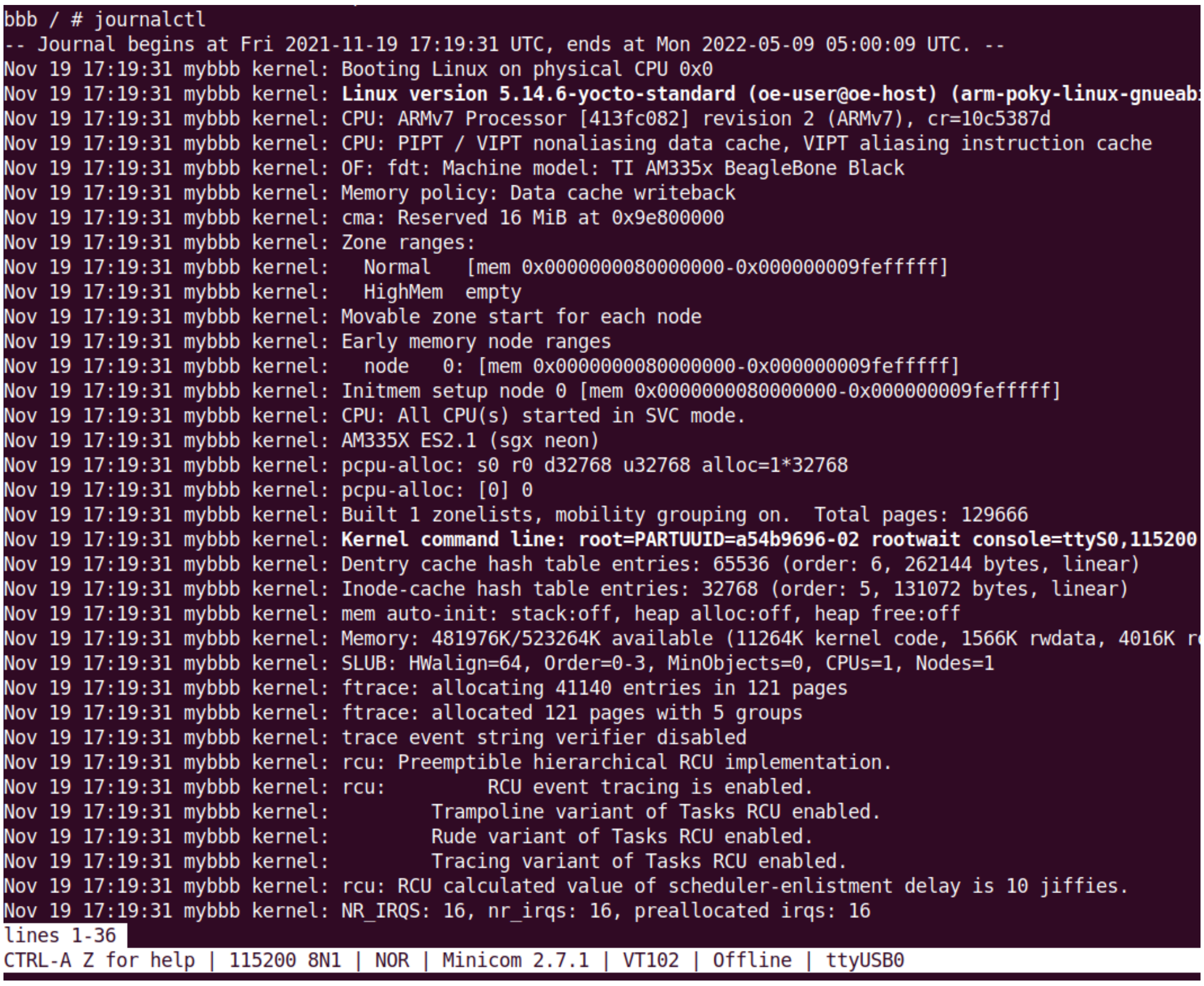
```
kernel/sched/core.c
   4601                struct task_struct *tsk = current;
   4602
   4603                sched_submit_work(tsk);
   4604                do {
   4605                        preempt_disable();
   4606                        __schedule(false);
   4607                        sched_preempt_enable_no_resched();
  >4608                } while (need_resched());
   4609                sched_update_worker(tsk);
   4610        }
   4611        EXPORT_SYMBOL(schedule);
   4612
   4613        /*
   4614         * synchronize_rcu_tasks() makes sure that no task is stuck in preempted
   4615         * state (have scheduled out non-voluntarily) by making sure that all
   4616         * tasks have either left the run queue or have gone into user space.
```

```
remote Thread 1.2 In: schedule                                                   L4608 PC: 0xffffffff833c9821
(gdb) hbreak schedule
Hardware assisted breakpoint 3 at 0xffffffff833c9770: file kernel/sched/core.c, line 4600.
(gdb) hbreak panic
Hardware assisted breakpoint 4 at 0xffffffff832ebe6e: file kernel/panic.c, line 178.
(gdb) i b
Num     Type           Disp Enb Address            What
3       hw breakpoint  keep y   0xffffffff833c9770 in schedule at kernel/sched/core.c:4600
4       hw breakpoint  keep y   0xffffffff832ebe6e in panic at kernel/panic.c:178
(gdb) c
Continuing.
[Switching to Thread 1.2]

Thread 2 hit Breakpoint 3, schedule () at kernel/sched/core.c:4600
(gdb) n
(gdb) p tsk
$1 = (struct task_struct *) 0xffff8880068d9780
(gdb) 
```

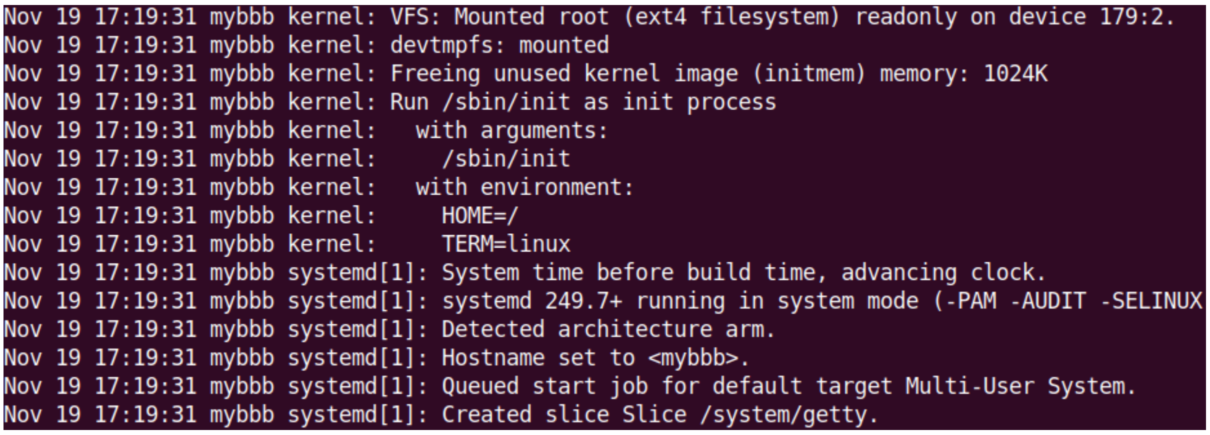
```
(gdb) help watch
Set a watchpoint for an expression.
Usage: watch [-l|-location] EXPRESSION
A watchpoint stops execution of your program whenever the value of
an expression changes.
If -l or -location is given, this evaluates EXPRESSION and watches
the memory to which it refers.
(gdb) watch jiffies_64
Hardware watchpoint 3: jiffies_64
(gdb) i b
Num     Type           Disp Enb Address            What
1       hw breakpoint  keep y   0xffffffff8299df54 in start_kernel at init/main.c:850
        breakpoint already hit 1 time
2       hw breakpoint  keep y   0xffffffff81ad87f7 in panic at kernel/panic.c:178
3       hw watchpoint  keep y                      jiffies_64
(gdb) c
Continuing.

Thread 1 hit Hardware watchpoint 3: jiffies_64

Old value = 4294892296
New value = 4294892297
do_timer (ticks=ticks@entry=1) at kernel/time/timekeeping.c:2269
2269            calc_global_load();
(gdb) bt
#0  do_timer (ticks=ticks@entry=1) at kernel/time/timekeeping.c:2269
#1  0xffffffff8119dd32 in tick_periodic (cpu=cpu@entry=0) at kernel/time/tick-common.c:93
#2  0xffffffff8119dd75 in tick_handle_periodic (dev=0xffff888003451400)
    at kernel/time/tick-common.c:111
#3  0xffffffff8108e808 in timer_interrupt (irq=<optimized out>, dev_id=<optimized out>)
    at arch/x86/kernel/time.c:57
#4  0xffffffff8116b465 in __handle_irq_event_percpu (desc=desc@entry=0xffff88800352c800,
    flags=flags@entry=0xffffc90000003f84) at kernel/irq/handle.c:156
#5  0xffffffff8116b5c3 in handle_irq_event_percpu (desc=desc@entry=0xffff88800352c800)
    at kernel/irq/handle.c:196
#6  0xffffffff8116b64b in handle_irq_event (desc=desc@entry=0xffff88800352c800)
    at kernel/irq/handle.c:213
```

Keyboard shortcut or command	**TUI mode action**
`Ctrl-x 2`	Cycle the content of the horizontally tiled windows – between the CPU registers view, source/assembly code view, and GDB command prompt.
`Ctrl-p`	Recall the previous command (the up arrow scrolls content in the focus window).
`Ctrl-n`	Recall the next command (the down arrow scrolls content in the focus window).
`fs next`	Switch the keyboard focus to the next window pane (useful to scroll content via the arrow keys in a pane).
`tui <cmd>`	`<cmd>` is one of `enable`, `disable`, or `reg`. `enable` and `disable` are self-explanatory; `reg` is explained next.
`tui reg <tab><tab>`	Display all possible CPU register display modes; the output is arch-dependent. On the x86_64, we get `all  float  general  mmx  next  prev  restore  save  sse  system  vector`.
`winheight`	Adjust the specified window's height (format: `winheight WINDOW-NAME [+ | -] NUM-LINES`).

Chapter 12: A Few More Kernel Debugging Approaches



```
bbb / # journalctl
-- Journal begins at Fri 2021-11-19 17:19:31 UTC, ends at Mon 2022-05-09 05:00:09 UTC. --
Nov 19 17:19:31 mybbb kernel: Booting Linux on physical CPU 0x0
Nov 19 17:19:31 mybbb kernel: Linux version 5.14.6-yocto-standard (oe-user@oe-host) (arm-poky-linux-gnueab
Nov 19 17:19:31 mybbb kernel: CPU: ARMv7 Processor [413fc082] revision 2 (ARMv7), cr=10c5387d
Nov 19 17:19:31 mybbb kernel: CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing instruction cache
Nov 19 17:19:31 mybbb kernel: OF: fdt: Machine model: TI AM335x BeagleBone Black
Nov 19 17:19:31 mybbb kernel: Memory policy: Data cache writeback
Nov 19 17:19:31 mybbb kernel: cma: Reserved 16 MiB at 0x9e800000
Nov 19 17:19:31 mybbb kernel: Zone ranges:
Nov 19 17:19:31 mybbb kernel:   Normal   [mem 0x0000000080000000-0x000000009fefffff]
Nov 19 17:19:31 mybbb kernel:   HighMem  empty
Nov 19 17:19:31 mybbb kernel: Movable zone start for each node
Nov 19 17:19:31 mybbb kernel: Early memory node ranges
Nov 19 17:19:31 mybbb kernel:   node   0: [mem 0x0000000080000000-0x000000009fefffff]
Nov 19 17:19:31 mybbb kernel: Initmem setup node 0 [mem 0x0000000080000000-0x000000009fefffff]
Nov 19 17:19:31 mybbb kernel: CPU: All CPU(s) started in SVC mode.
Nov 19 17:19:31 mybbb kernel: AM335X ES2.1 (sgx neon)
Nov 19 17:19:31 mybbb kernel: pcpu-alloc: s0 r0 d32768 u32768 alloc=1*32768
Nov 19 17:19:31 mybbb kernel: pcpu-alloc: [0] 0
Nov 19 17:19:31 mybbb kernel: Built 1 zonelists, mobility grouping on.  Total pages: 129666
Nov 19 17:19:31 mybbb kernel: Kernel command line: root=PARTUUID=a54b9696-02 rootwait console=ttyS0,115200
Nov 19 17:19:31 mybbb kernel: Dentry cache hash table entries: 65536 (order: 6, 262144 bytes, linear)
Nov 19 17:19:31 mybbb kernel: Inode-cache hash table entries: 32768 (order: 5, 131072 bytes, linear)
Nov 19 17:19:31 mybbb kernel: mem auto-init: stack:off, heap alloc:off, heap free:off
Nov 19 17:19:31 mybbb kernel: Memory: 481976K/523264K available (11264K kernel code, 1566K rwdata, 4016K r
Nov 19 17:19:31 mybbb kernel: SLUB: HWalign=64, Order=0-3, MinObjects=0, CPUs=1, Nodes=1
Nov 19 17:19:31 mybbb kernel: ftrace: allocating 41140 entries in 121 pages
Nov 19 17:19:31 mybbb kernel: ftrace: allocated 121 pages with 5 groups
Nov 19 17:19:31 mybbb kernel: trace event string verifier disabled
Nov 19 17:19:31 mybbb kernel: rcu: Preemptible hierarchical RCU implementation.
Nov 19 17:19:31 mybbb kernel: rcu:         RCU event tracing is enabled.
Nov 19 17:19:31 mybbb kernel:         Trampoline variant of Tasks RCU enabled.
Nov 19 17:19:31 mybbb kernel:         Rude variant of Tasks RCU enabled.
Nov 19 17:19:31 mybbb kernel:         Tracing variant of Tasks RCU enabled.
Nov 19 17:19:31 mybbb kernel: rcu: RCU calculated value of scheduler-enlistment delay is 10 jiffies.
Nov 19 17:19:31 mybbb kernel: NR_IRQS: 16, nr_irqs: 16, preallocated irqs: 16
lines 1-36
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Offline | ttyUSB0
```



```
Nov 19 17:19:31 mybbb kernel: VFS: Mounted root (ext4 filesystem) readonly on device 179:2.
Nov 19 17:19:31 mybbb kernel: devtmpfs: mounted
Nov 19 17:19:31 mybbb kernel: Freeing unused kernel image (initmem) memory: 1024K
Nov 19 17:19:31 mybbb kernel: Run /sbin/init as init process
Nov 19 17:19:31 mybbb kernel:   with arguments:
Nov 19 17:19:31 mybbb kernel:     /sbin/init
Nov 19 17:19:31 mybbb kernel:   with environment:
Nov 19 17:19:31 mybbb kernel:     HOME=/
Nov 19 17:19:31 mybbb kernel:     TERM=linux
Nov 19 17:19:31 mybbb systemd[1]: System time before build time, advancing clock.
Nov 19 17:19:31 mybbb systemd[1]: systemd 249.7+ running in system mode (-PAM -AUDIT -SELINUX
Nov 19 17:19:31 mybbb systemd[1]: Detected architecture arm.
Nov 19 17:19:31 mybbb systemd[1]: Hostname set to <mybbb>.
Nov 19 17:19:31 mybbb systemd[1]: Queued start job for default target Multi-User System.
Nov 19 17:19:31 mybbb systemd[1]: Created slice Slice /system/getty.
```