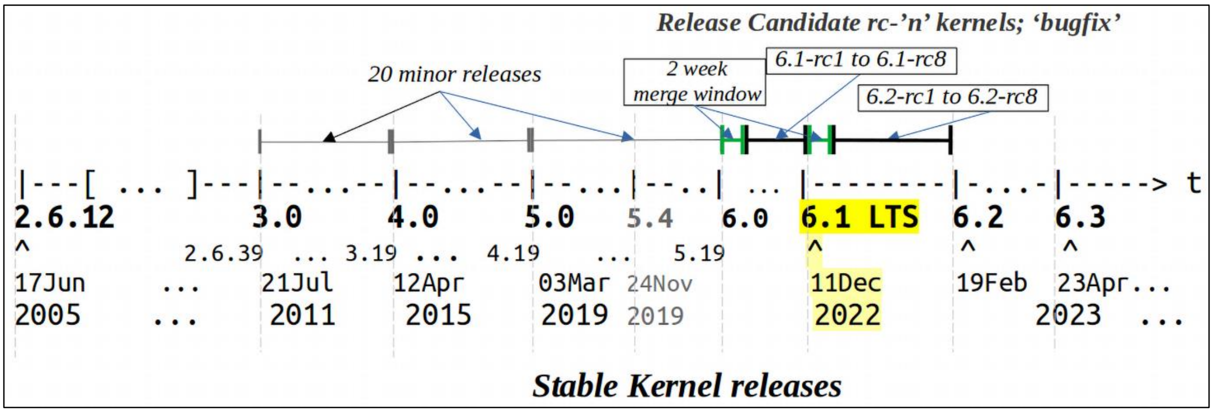
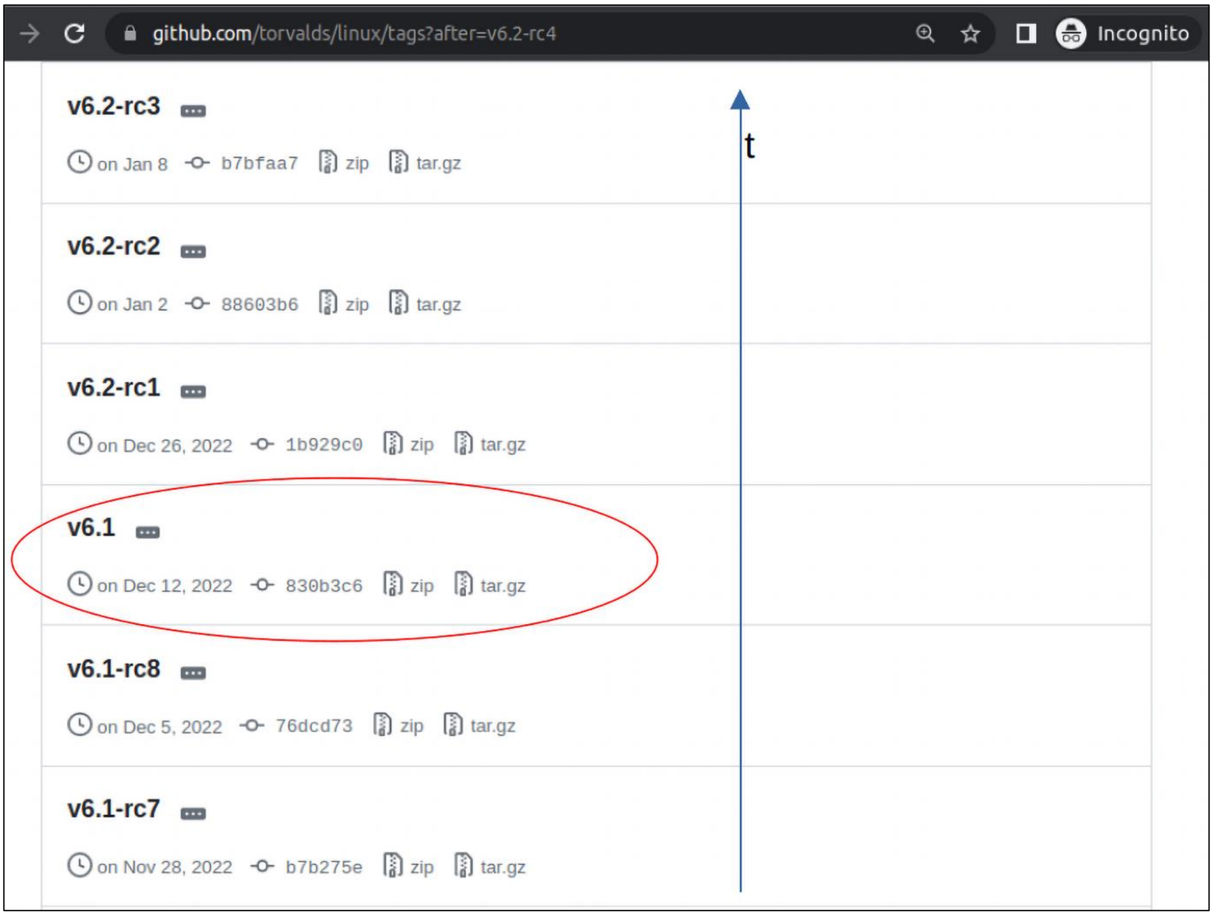
Chapter 2: Building the 6.x Linux Kernel from Source - Part 1




github.com/torvalds/linux/tags?after=v6.2-rc4

v6.2-rc3
on Jan 8 b7bfaa7 zip tar.gz

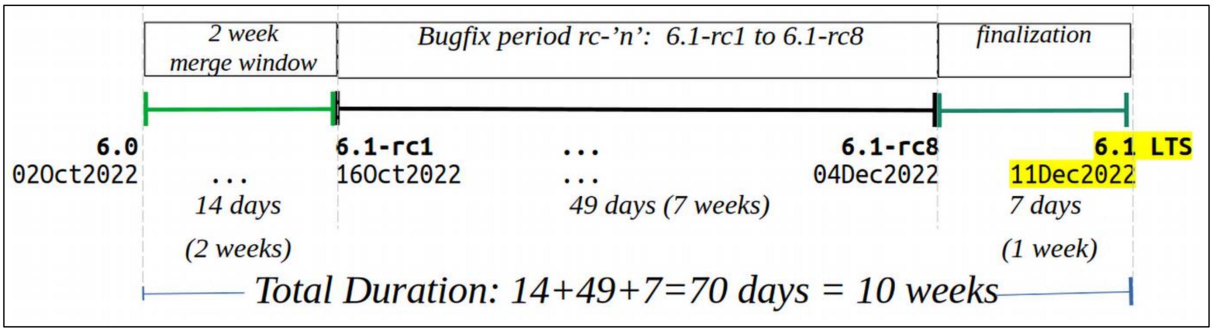
v6.2-rc2
on Jan 2 88603b6 zip tar.gz

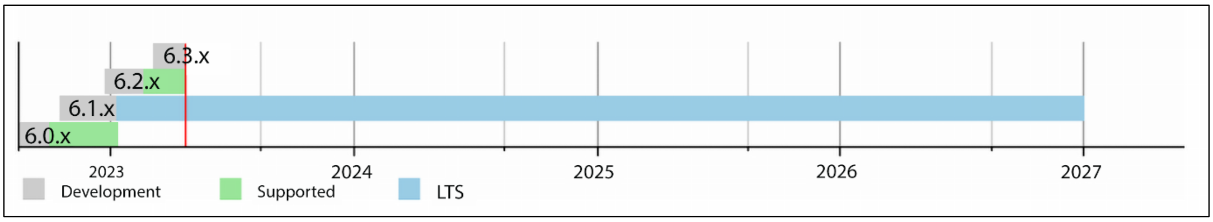
v6.2-rc1
on Dec 26, 2022 1b929c0 zip tar.gz

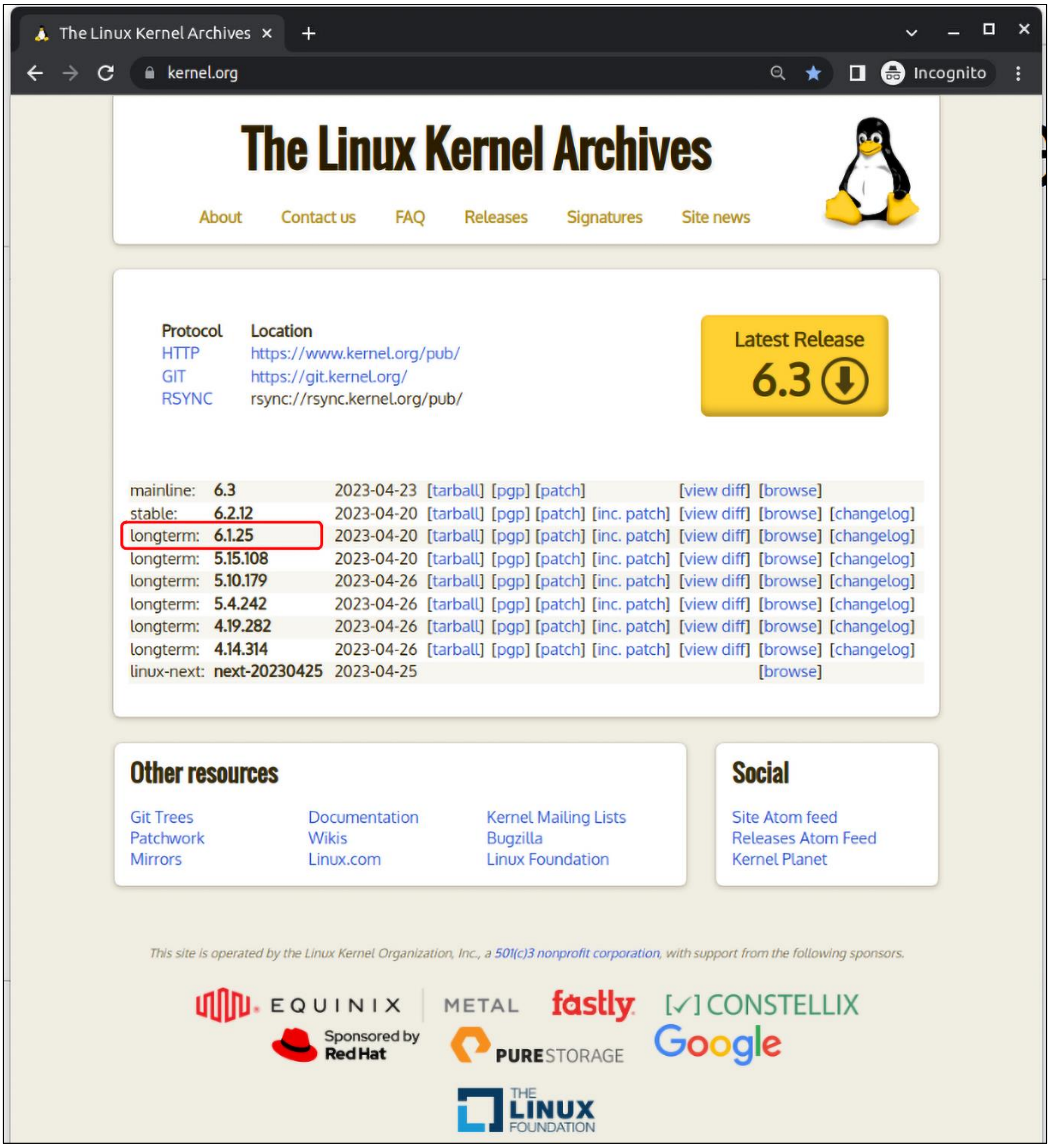
v6.1
on Dec 12, 2022 830b3c6 zip tar.gz

v6.1-rc8
on Dec 5, 2022 76dcd73 zip tar.gz

v6.1-rc7
on Nov 28, 2022 b7b275e zip tar.gz


2 week merge window
Bugfix period rc-'n': 6.1-rc1 to 6.1-rc8
finalization
6.0
02Oct2022
...
14 days
(2 weeks)
6.1-rc1
16Oct2022
...
...
49 days (7 weeks)
6.1-rc8
04Dec2022
6.1 LTS
11Dec2022
7 days
(1 week)
Total Duration: 14+49+7=70 days = 10 weeks


6.3.x
6.2.x
6.1.x
6.0.x
2023
2024
2025
2026
2027
Development
Supported
LTS



The Linux Kernel Archives

About Contact us FAQ Releases Signatures Site news

Protocol	Location
HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

Latest Release
6.3

mainline:	6.3	2023-04-23	[tarball] [pgp] [patch]		[view diff]	[browse]	
stable:	6.2.12	2023-04-20	[tarball] [pgp] [patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	6.1.25	2023-04-20	[tarball] [pgp] [patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.15.108	2023-04-20	[tarball] [pgp] [patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.10.179	2023-04-26	[tarball] [pgp] [patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.4.242	2023-04-26	[tarball] [pgp] [patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.19.282	2023-04-26	[tarball] [pgp] [patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.14.314	2023-04-26	[tarball] [pgp] [patch]	[inc. patch]	[view diff]	[browse]	[changelog]
linux-next:	next-20230425	2023-04-25				[browse]	

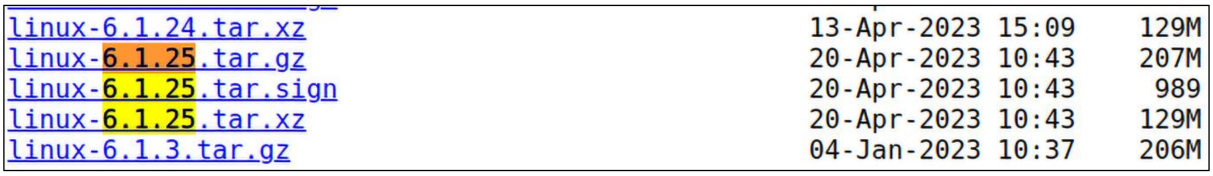
Other resources

Git Trees
Patchwork
Mirrors
Documentation
Wikis
Linux.com
Kernel Mailing Lists
Bugzilla
Linux Foundation

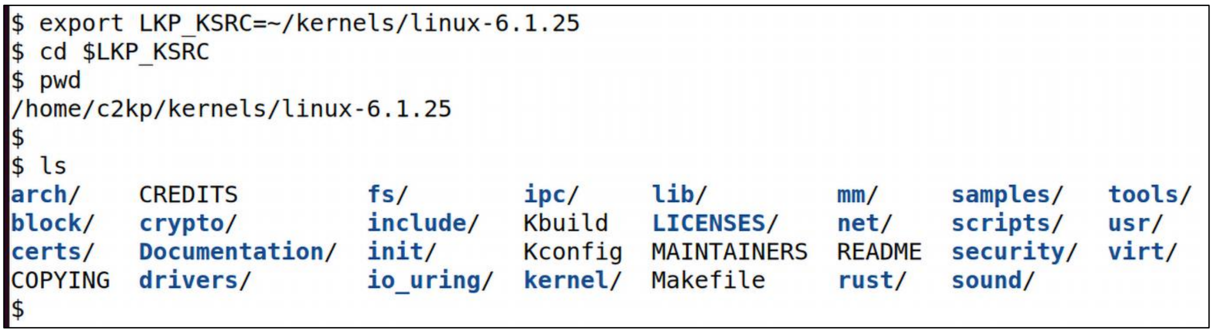
Social

Site Atom feed
Releases Atom Feed
Kernel Planet

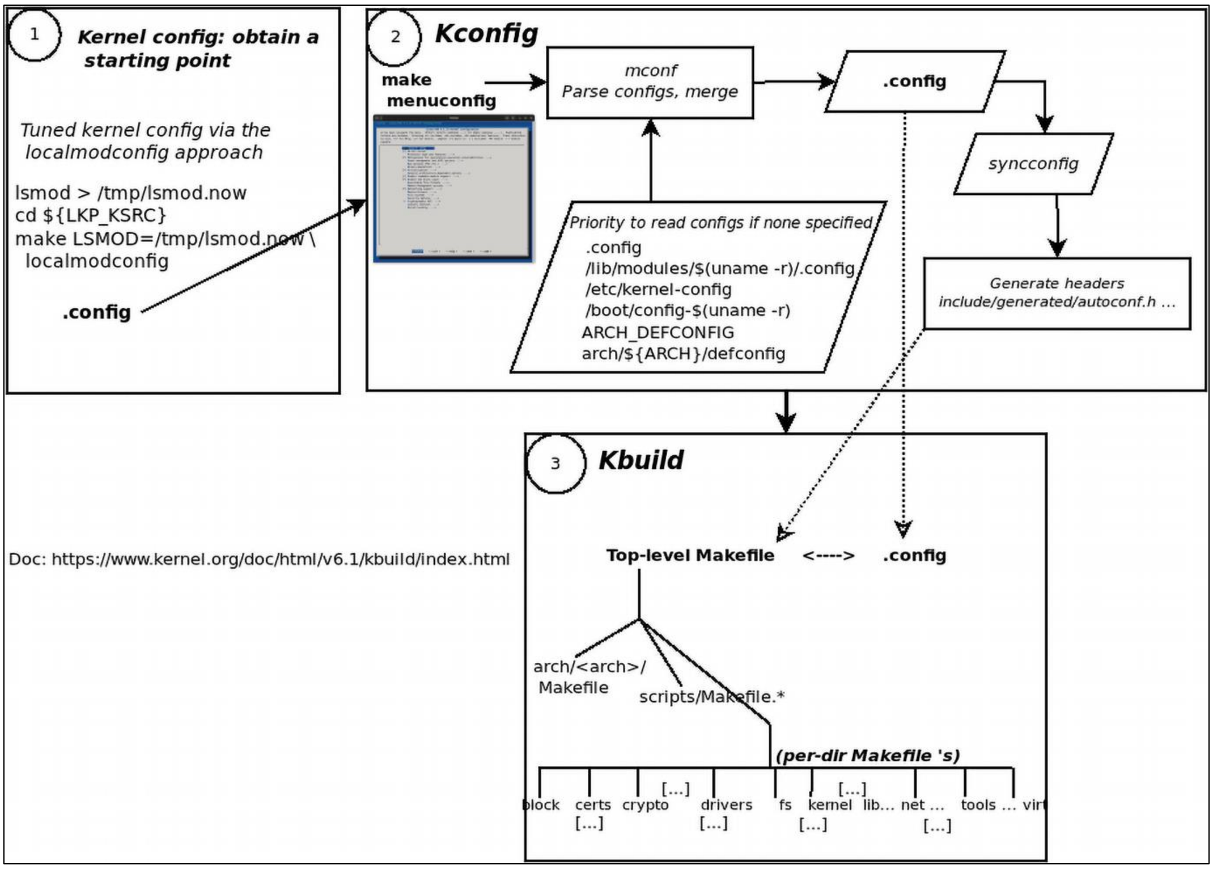
This site is operated by the Linux Kernel Organization, Inc., a 501(c)3 nonprofit corporation, with support from the following sponsors.

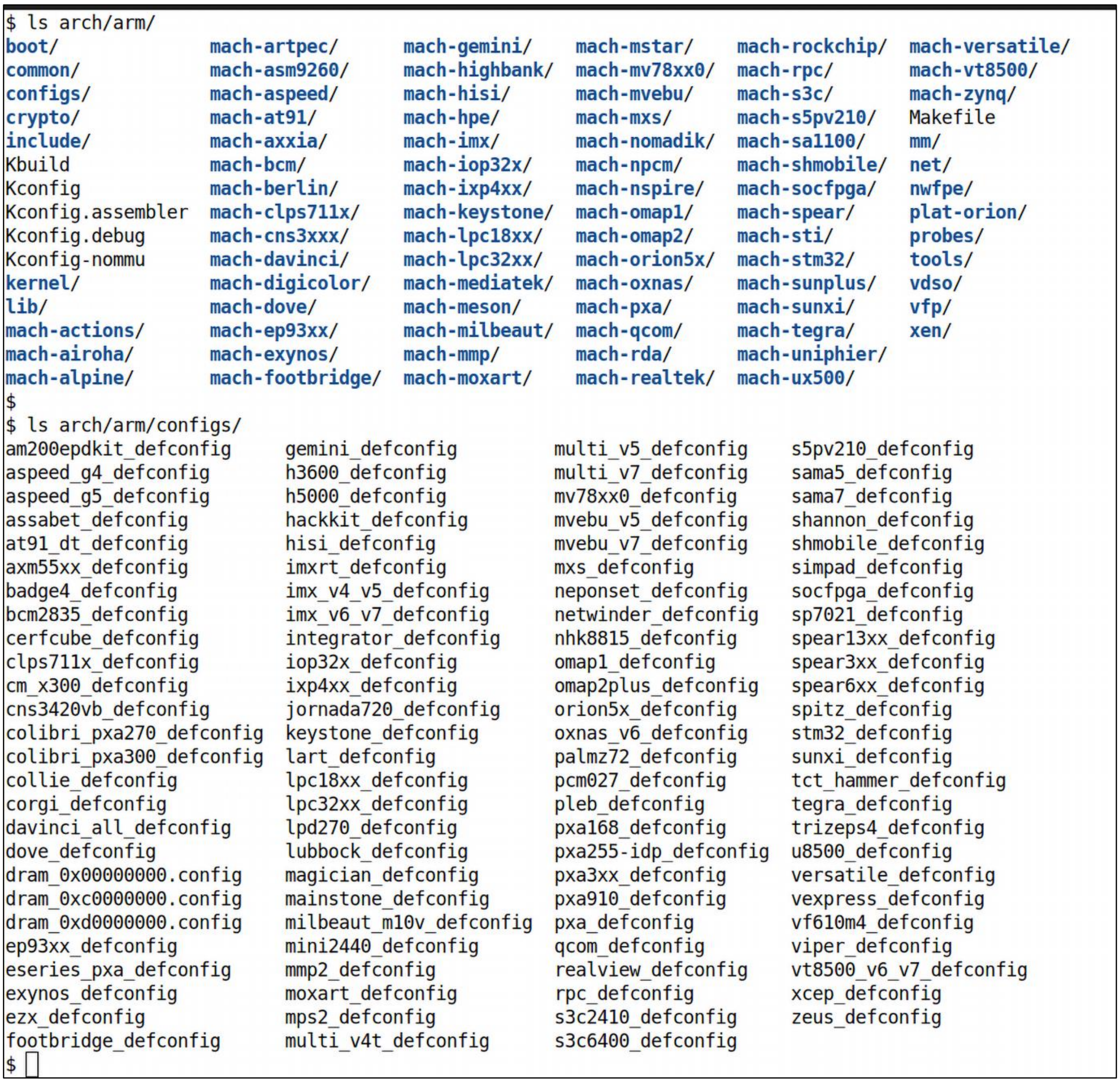


```
linux-6.1.24.tar.xz                         13-Apr-2023 15:09     129M
linux-6.1.25.tar.gz                         20-Apr-2023 10:43     207M
linux-6.1.25.tar.sign                       20-Apr-2023 10:43      989
linux-6.1.25.tar.xz                         20-Apr-2023 10:43     129M
linux-6.1.3.tar.gz                          04-Jan-2023 10:37     206M
```

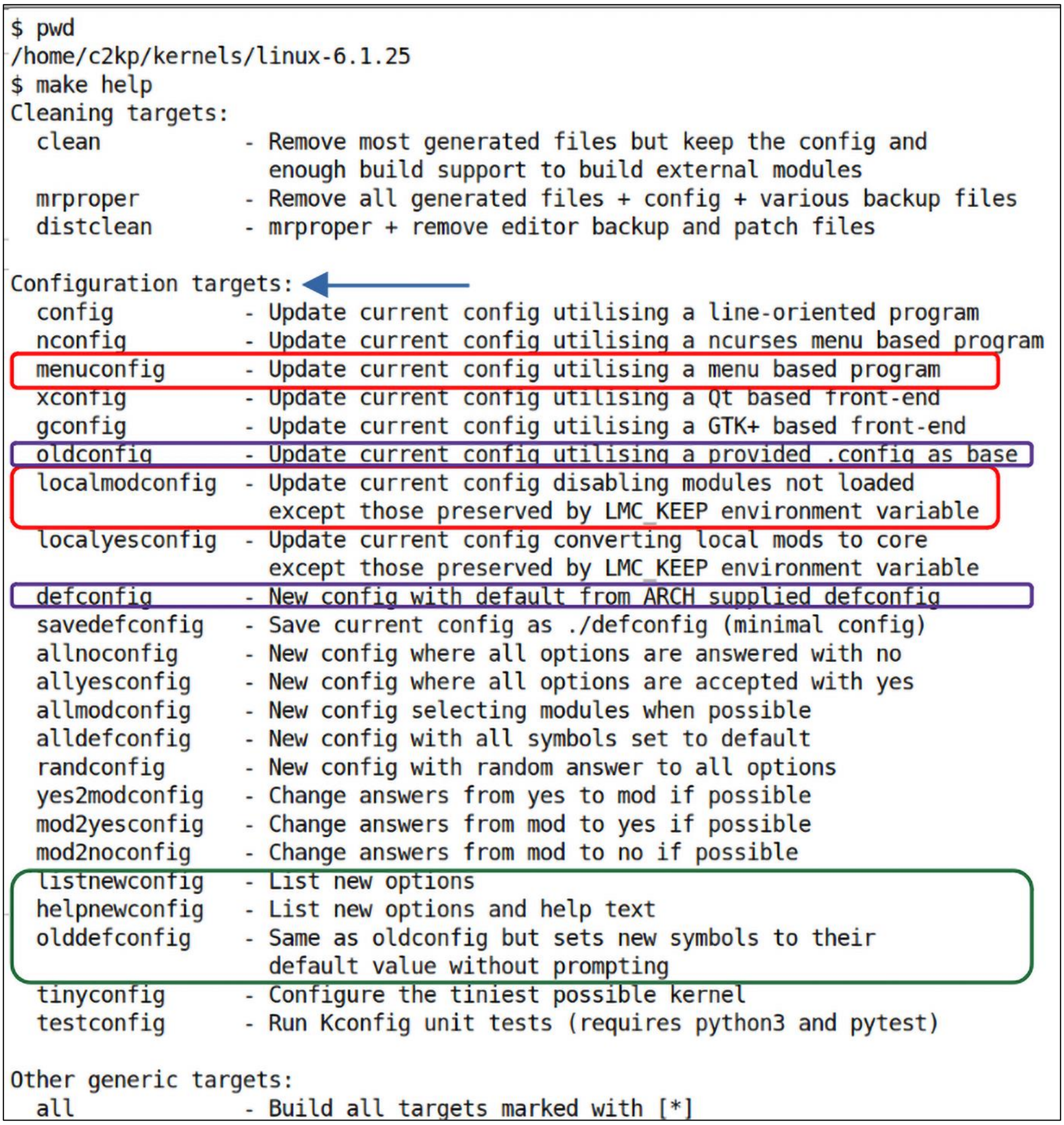


```
$ export LKP_KSRC=~/kernels/linux-6.1.25
$ cd $LKP_KSRC
$ pwd
/home/c2kp/kernels/linux-6.1.25
$
$ ls
arch/    CREDITS   fs/        ipc/     lib/         mm/     samples/   tools/
block/   crypto/   include/   Kbuild   LICENSES/    net/    scripts/   usr/
certs/   Documentation/  init/  Kconfig  MAINTAINERS  README  security/  virt/
COPYING  drivers/  io_uring/  kernel/  Makefile     rust/   sound/
$
```

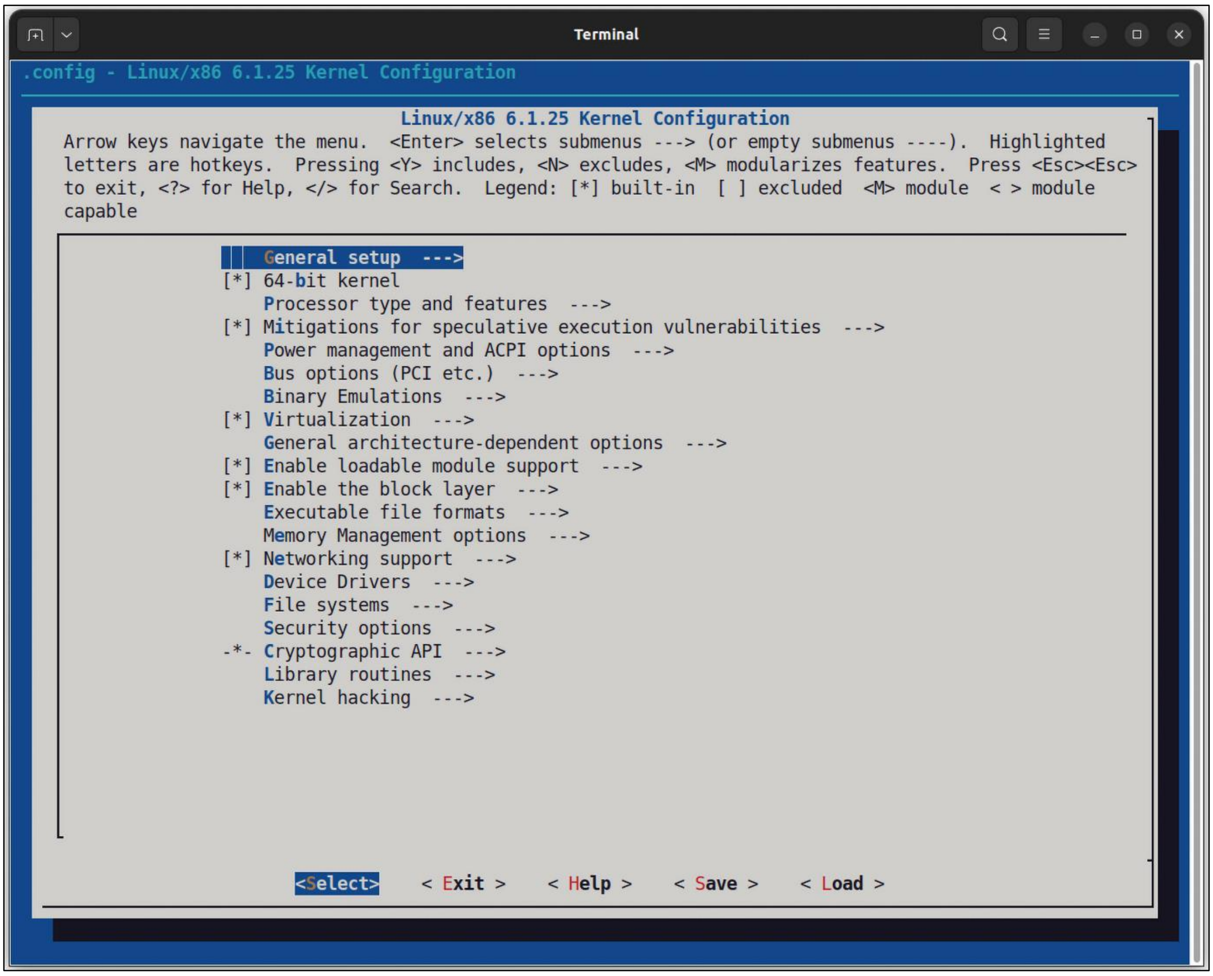




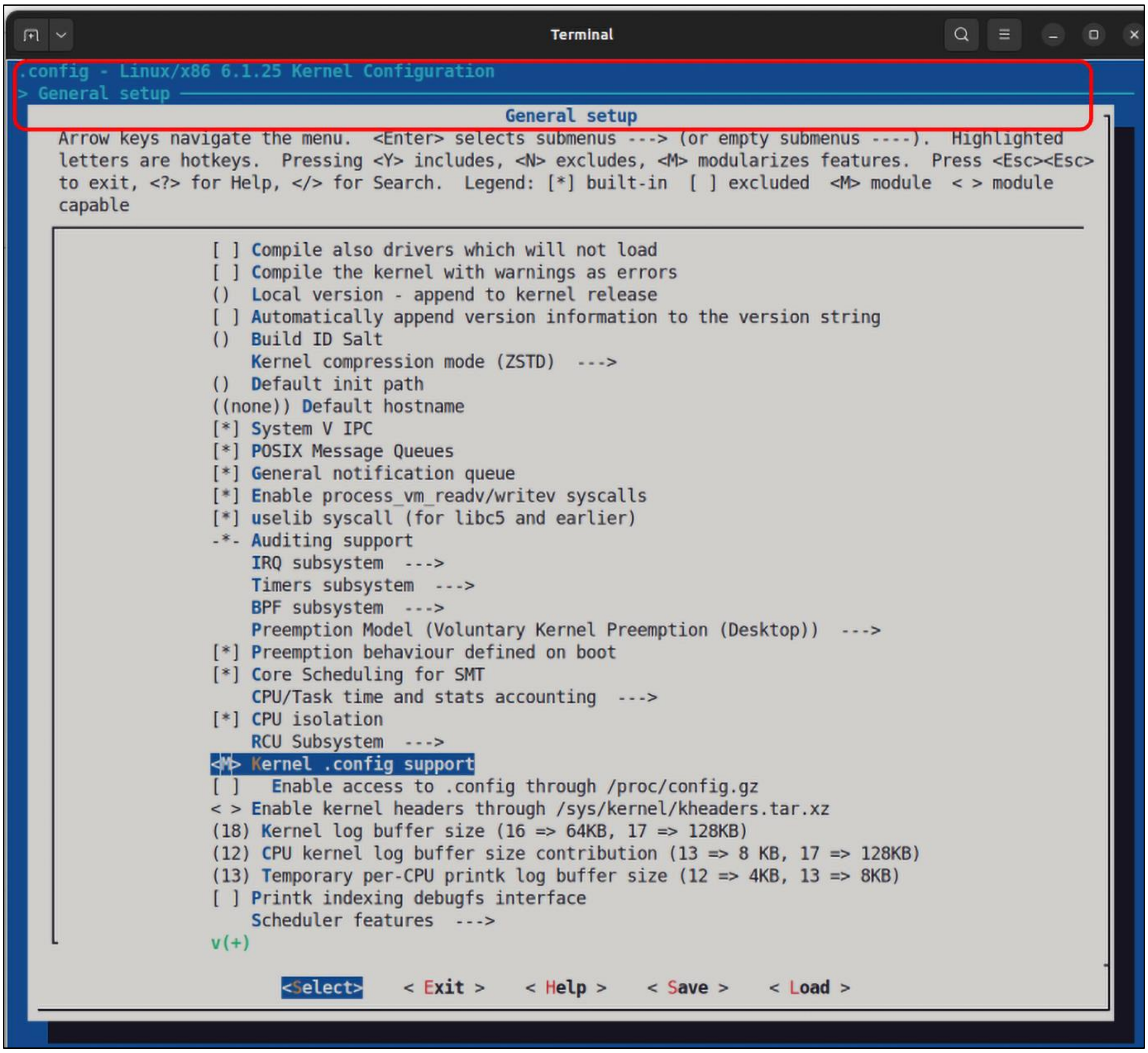
```
$ ls arch/arm/
boot/              mach-artpec/      mach-gemini/    mach-mstar/    mach-rockchip/  mach-versatile/
common/            mach-asm9260/     mach-highbank/  mach-mv78xx0/  mach-rpc/       mach-vt8500/
configs/           mach-aspeed/      mach-hisi/      mach-mvebu/    mach-s3c/       mach-zynq/
crypto/            mach-at91/        mach-hpe/       mach-mxs/      mach-s5pv210/   Makefile
include/           mach-axxia/       mach-imx/       mach-nomadik/  mach-sa1100/    mm/
Kbuild             mach-bcm/         mach-iop32x/    mach-npcm/     mach-shmobile/  net/
Kconfig            mach-berlin/      mach-ixp4xx/    mach-nspire/   mach-socfpga/   nwfpe/
Kconfig.assembler  mach-clps711x/    mach-keystone/  mach-omap1/    mach-spear/     plat-orion/
Kconfig.debug      mach-cns3xxx/     mach-lpc18xx/   mach-omap2/    mach-sti/       probes/
Kconfig-nommu      mach-davinci/     mach-lpc32xx/   mach-orion5x/  mach-stm32/     tools/
kernel/            mach-digicolor/   mach-mediatek/  mach-oxnas/    mach-sunplus/   vdso/
lib/               mach-dove/        mach-meson/     mach-pxa/      mach-sunxi/     vfp/
mach-actions/      mach-ep93xx/      mach-milbeaut/  mach-qcom/     mach-tegra/     xen/
mach-airoha/       mach-exynos/      mach-mmp/       mach-rda/      mach-uniphier/
mach-alpine/       mach-footbridge/  mach-moxart/    mach-realtek/  mach-ux500/
$
$ ls arch/arm/configs/
am200epdkit_defconfig     gemini_defconfig         multi_v5_defconfig    s5pv210_defconfig
aspeed_g4_defconfig       h3600_defconfig          multi_v7_defconfig    sama5_defconfig
aspeed_g5_defconfig       h5000_defconfig          mv78xx0_defconfig     sama7_defconfig
assabet_defconfig         hackkit_defconfig        mvebu_v5_defconfig    shannon_defconfig
at91_dt_defconfig         hisi_defconfig           mvebu_v7_defconfig    shmobile_defconfig
axm55xx_defconfig         imxrt_defconfig          mxs_defconfig         simpad_defconfig
badge4_defconfig          imx_v4_v5_defconfig      neponset_defconfig    socfpga_defconfig
bcm2835_defconfig         imx_v6_v7_defconfig      netwinder_defconfig   sp7021_defconfig
cerfcube_defconfig        integrator_defconfig     nhk8815_defconfig     spear13xx_defconfig
clps711x_defconfig        iop32x_defconfig         omap1_defconfig       spear3xx_defconfig
cm_x300_defconfig         ixp4xx_defconfig         omap2plus_defconfig   spear6xx_defconfig
cns3420vb_defconfig       jornada720_defconfig     orion5x_defconfig     spitz_defconfig
colibri_pxa270_defconfig  keystone_defconfig       oxnas_v6_defconfig    stm32_defconfig
colibri_pxa300_defconfig  lart_defconfig           palmz72_defconfig     sunxi_defconfig
collie_defconfig          lpc18xx_defconfig        pcm027_defconfig      tct_hammer_defconfig
corgi_defconfig           lpc32xx_defconfig        pleb_defconfig        tegra_defconfig
davinci_all_defconfig     lpd270_defconfig         pxa168_defconfig      trizeps4_defconfig
dove_defconfig            lubbock_defconfig        pxa255-idp_defconfig  u8500_defconfig
dram_0x00000000.config    magician_defconfig       pxa3xx_defconfig      versatile_defconfig
dram_0xc0000000.config    mainstone_defconfig      pxa910_defconfig      vexpress_defconfig
dram_0xd0000000.config    milbeaut_m10v_defconfig  pxa_defconfig         vf610m4_defconfig
ep93xx_defconfig          mini2440_defconfig       qcom_defconfig        viper_defconfig
eseries_pxa_defconfig     mmp2_defconfig           realview_defconfig    vt8500_v6_v7_defconfig
exynos_defconfig          moxart_defconfig         rpc_defconfig         xcep_defconfig
ezx_defconfig             mps2_defconfig           s3c2410_defconfig     zeus_defconfig
footbridge_defconfig      multi_v4t_defconfig      s3c6400_defconfig
$
```



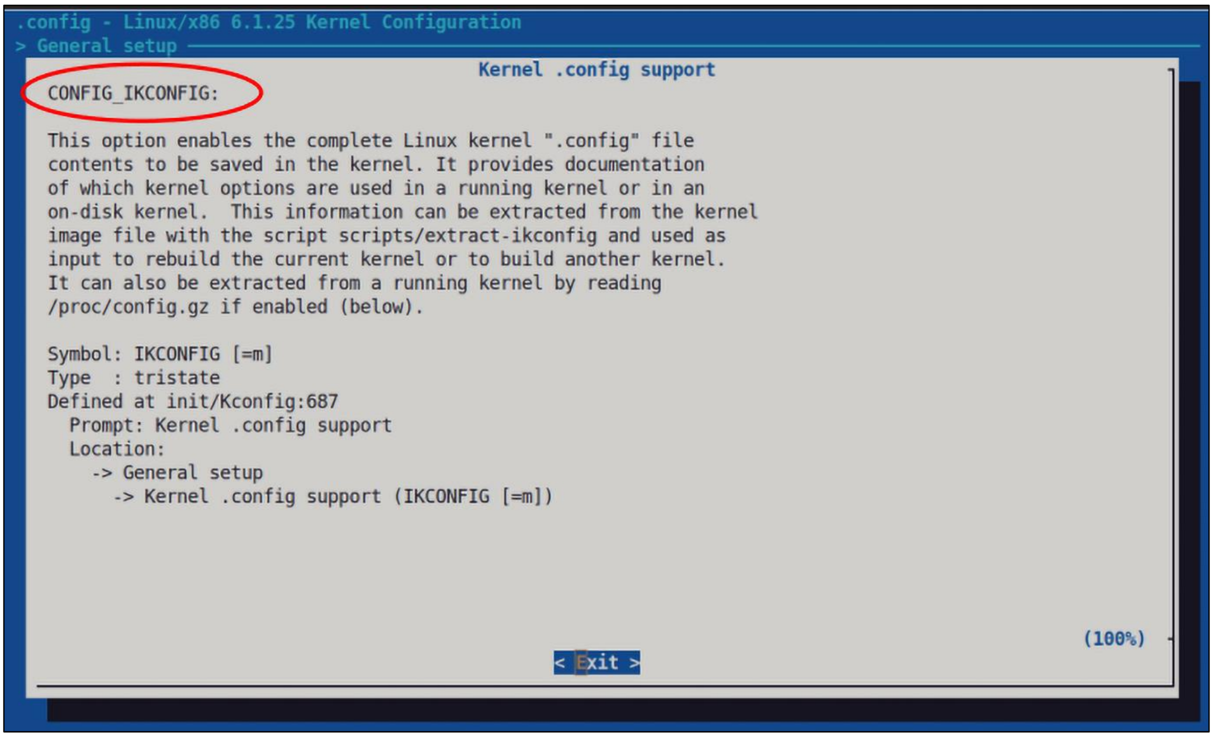
```
$ pwd
/home/c2kp/kernels/linux-6.1.25
$ make help
Cleaning targets:
  clean           - Remove most generated files but keep the config and
                    enough build support to build external modules
  mrproper        - Remove all generated files + config + various backup files
  distclean       - mrproper + remove editor backup and patch files

Configuration targets:
  config          - Update current config utilising a line-oriented program
  nconfig         - Update current config utilising a ncurses menu based program
  menuconfig      - Update current config utilising a menu based program
  xconfig         - Update current config utilising a Qt based front-end
  gconfig         - Update current config utilising a GTK+ based front-end
  oldconfig       - Update current config utilising a provided .config as base
  localmodconfig  - Update current config disabling modules not loaded
                    except those preserved by LMC_KEEP environment variable
  localyesconfig  - Update current config converting local mods to core
                    except those preserved by LMC_KEEP environment variable
  defconfig       - New config with default from ARCH supplied defconfig
  savedefconfig   - Save current config as ./defconfig (minimal config)
  allnoconfig     - New config where all options are answered with no
  allyesconfig    - New config where all options are accepted with yes
  allmodconfig    - New config selecting modules when possible
  alldefconfig    - New config with all symbols set to default
  randconfig      - New config with random answer to all options
  yes2modconfig   - Change answers from yes to mod if possible
  mod2yesconfig   - Change answers from mod to yes if possible
  mod2noconfig    - Change answers from mod to no if possible
  listnewconfig   - List new options
  helpnewconfig   - List new options and help text
  olddefconfig    - Same as oldconfig but sets new symbols to their
                    default value without prompting
  tinyconfig      - Configure the tiniest possible kernel
  testconfig      - Run Kconfig unit tests (requires python3 and pytest)

Other generic targets:
  all             - Build all targets marked with [*]
```



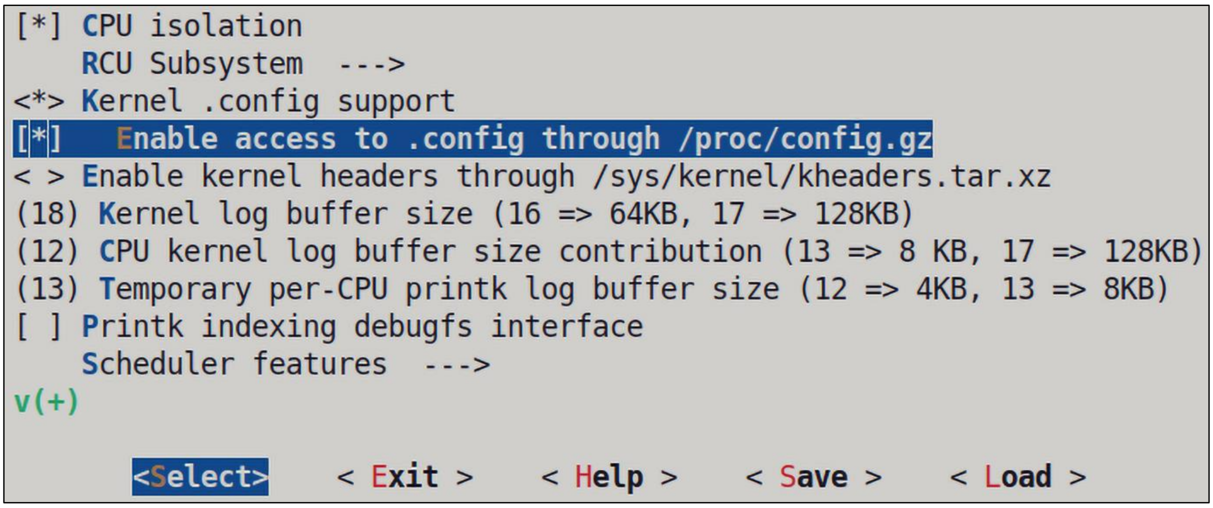
```
.config - Linux/x86 6.1.25 Kernel Configuration

                    Linux/x86 6.1.25 Kernel Configuration
  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

                  General setup  --->
              [*] 64-bit kernel
                  Processor type and features  --->
              [*] Mitigations for speculative execution vulnerabilities  --->
                  Power management and ACPI options  --->
                  Bus options (PCI etc.)  --->
                  Binary Emulations  --->
              [*] Virtualization  --->
                  General architecture-dependent options  --->
              [*] Enable loadable module support  --->
              [*] Enable the block layer  --->
                  Executable file formats  --->
                  Memory Management options  --->
              [*] Networking support  --->
                  Device Drivers  --->
                  File systems  --->
                  Security options  --->
              -*- Cryptographic API  --->
                  Library routines  --->
                  Kernel hacking  --->

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



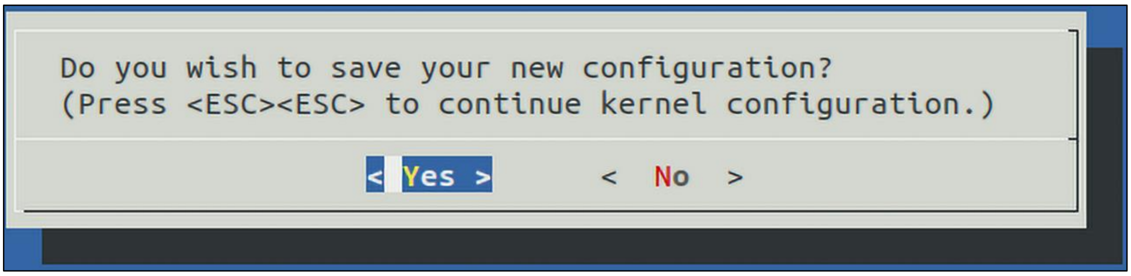
```
.config - Linux/x86 6.1.25 Kernel Configuration
> General setup ─────────────────────────────────────────────────────────
                              General setup
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted
  letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc>
  to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module
  capable

                [ ] Compile also drivers which will not load
                [ ] Compile the kernel with warnings as errors
                ()  Local version - append to kernel release
                [ ] Automatically append version information to the version string
                ()  Build ID Salt
                    Kernel compression mode (ZSTD)  --->
                ()  Default init path
                ((none)) Default hostname
                [*] System V IPC
                [*] POSIX Message Queues
                [*] General notification queue
                [*] Enable process_vm_readv/writev syscalls
                [*] uselib syscall (for libc5 and earlier)
                -*- Auditing support
                    IRQ subsystem  --->
                    Timers subsystem  --->
                    BPF subsystem  --->
                    Preemption Model (Voluntary Kernel Preemption (Desktop))  --->
                [*] Preemption behaviour defined on boot
                [*] Core Scheduling for SMT
                    CPU/Task time and stats accounting  --->
                [*] CPU isolation
                    RCU Subsystem  --->
                <M> Kernel .config support
                [ ]   Enable access to .config through /proc/config.gz
                < > Enable kernel headers through /sys/kernel/kheaders.tar.xz
                (18) Kernel log buffer size (16 => 64KB, 17 => 128KB)
                (12) CPU kernel log buffer size contribution (13 => 8 KB, 17 => 128KB)
                (13) Temporary per-CPU printk log buffer size (12 => 4KB, 13 => 8KB)
                [ ] Printk indexing debugfs interface
                    Scheduler features  --->
                v(+)

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



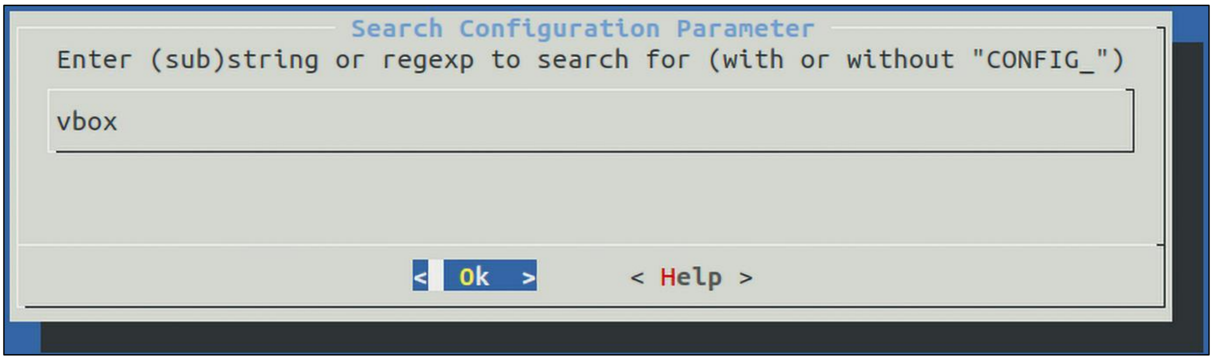
```
.config - Linux/x86 6.1.25 Kernel Configuration
> General setup ──────────────────────────────────────────────────
                          Kernel .config support
 CONFIG_IKCONFIG:

 This option enables the complete Linux kernel ".config" file
 contents to be saved in the kernel. It provides documentation
 of which kernel options are used in a running kernel or in an
 on-disk kernel.  This information can be extracted from the kernel
 image file with the script scripts/extract-ikconfig and used as
 input to rebuild the current kernel or to build another kernel.
 It can also be extracted from a running kernel by reading
 /proc/config.gz if enabled (below).

 Symbol: IKCONFIG [=m]
 Type  : tristate
 Defined at init/Kconfig:687
   Prompt: Kernel .config support
   Location:
     -> General setup
       -> Kernel .config support (IKCONFIG [=m])

                                                          (100%)
                             < Exit >
```



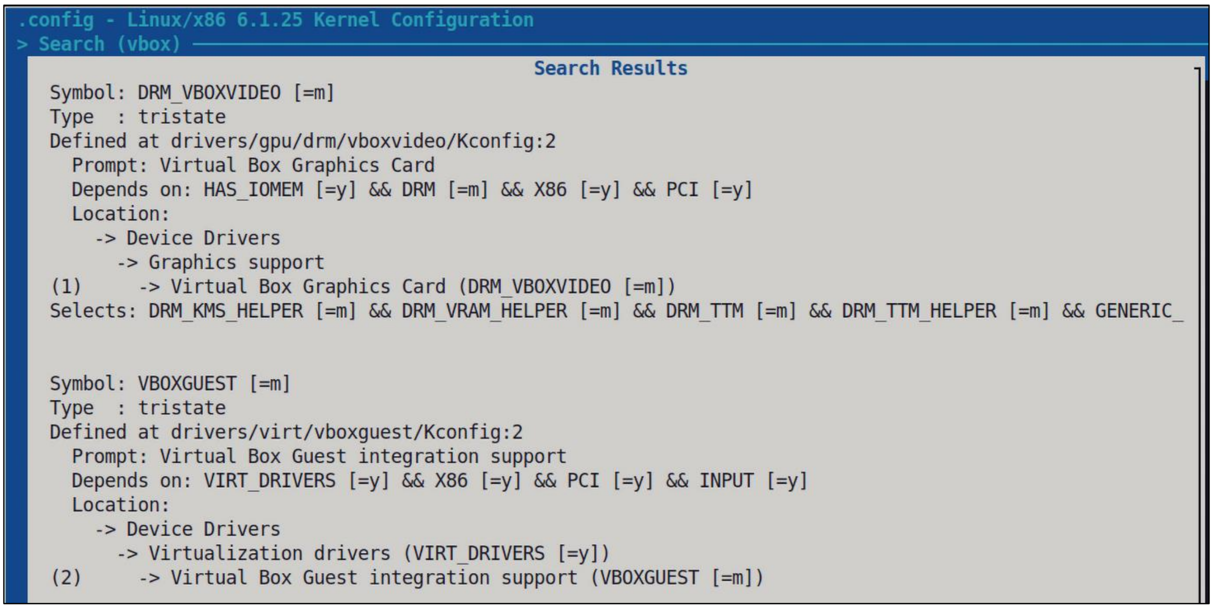
```
[*] CPU isolation
    RCU Subsystem  --->
<*> Kernel .config support
[*]   Enable access to .config through /proc/config.gz
< > Enable kernel headers through /sys/kernel/kheaders.tar.xz
(18) Kernel log buffer size (16 => 64KB, 17 => 128KB)
(12) CPU kernel log buffer size contribution (13 => 8 KB, 17 => 128KB)
(13) Temporary per-CPU printk log buffer size (12 => 4KB, 13 => 8KB)
[ ] Printk indexing debugfs interface
    Scheduler features  --->
v(+)

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



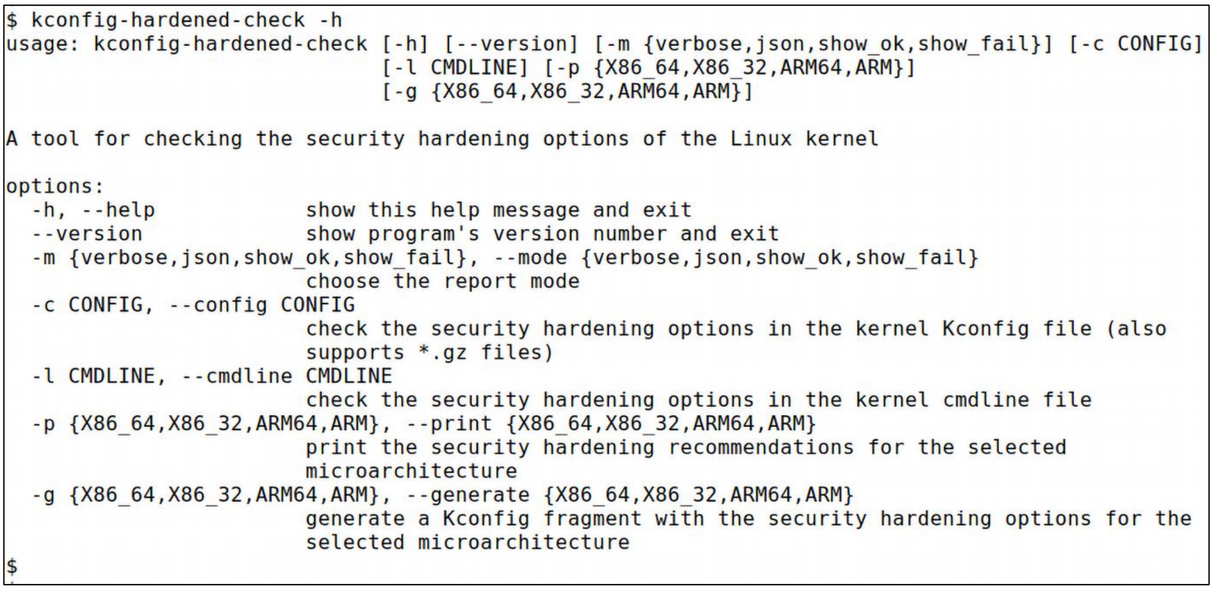
```
  Do you wish to save your new configuration?
  (Press <ESC><ESC> to continue kernel configuration.)

                 < Yes >        <  No  >
```



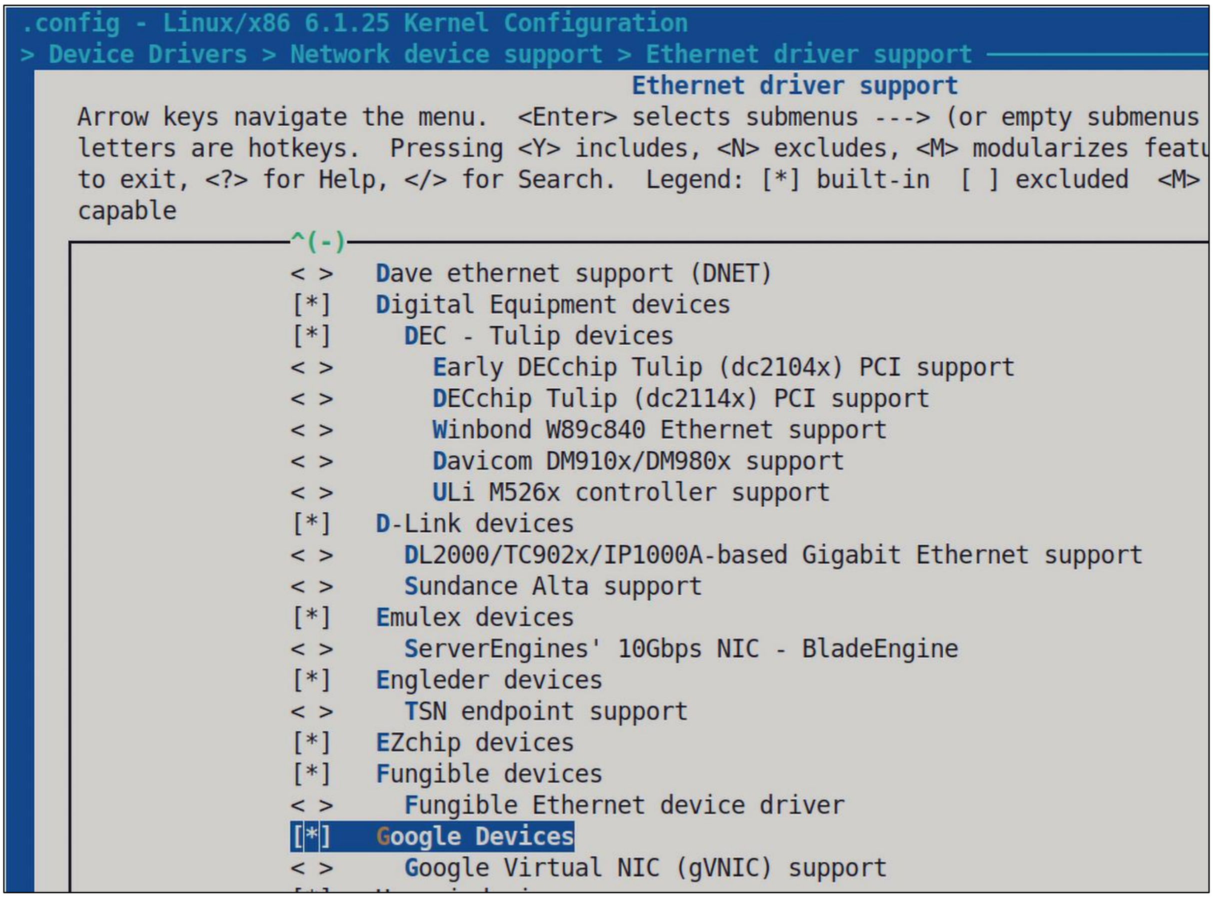
```
Search Configuration Parameter
Enter (sub)string or regexp to search for (with or without "CONFIG_")

vbox
```



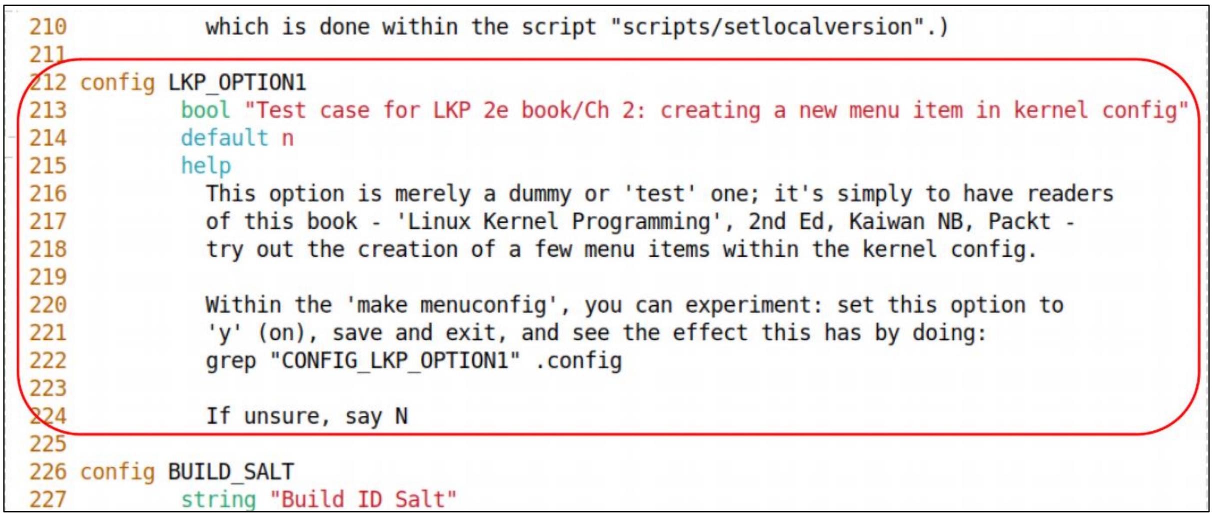
```
.config - Linux/x86 6.1.25 Kernel Configuration
> Search (vbox)
                                   Search Results
 Symbol: DRM_VBOXVIDEO [=m]
 Type  : tristate
 Defined at drivers/gpu/drm/vboxvideo/Kconfig:2
   Prompt: Virtual Box Graphics Card
   Depends on: HAS_IOMEM [=y] && DRM [=m] && X86 [=y] && PCI [=y]
   Location:
     -> Device Drivers
       -> Graphics support
 (1)     -> Virtual Box Graphics Card (DRM_VBOXVIDEO [=m])
 Selects: DRM_KMS_HELPER [=m] && DRM_VRAM_HELPER [=m] && DRM_TTM [=m] && DRM_TTM_HELPER [=m] && GENERIC_


 Symbol: VBOXGUEST [=m]
 Type  : tristate
 Defined at drivers/virt/vboxguest/Kconfig:2
   Prompt: Virtual Box Guest integration support
   Depends on: VIRT_DRIVERS [=y] && X86 [=y] && PCI [=y] && INPUT [=y]
   Location:
     -> Device Drivers
       -> Virtualization drivers (VIRT_DRIVERS [=y])
 (2)     -> Virtual Box Guest integration support (VBOXGUEST [=m])
```



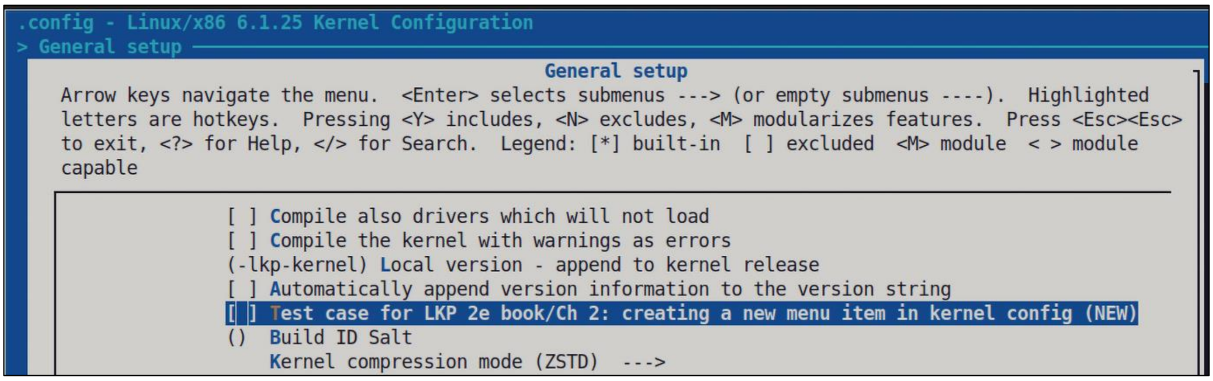
```
$ kconfig-hardened-check -h
usage: kconfig-hardened-check [-h] [--version] [-m {verbose,json,show_ok,show_fail}] [-c CONFIG]
                              [-l CMDLINE] [-p {X86_64,X86_32,ARM64,ARM}]
                              [-g {X86_64,X86_32,ARM64,ARM}]

A tool for checking the security hardening options of the Linux kernel

options:
  -h, --help            show this help message and exit
  --version             show program's version number and exit
  -m {verbose,json,show_ok,show_fail}, --mode {verbose,json,show_ok,show_fail}
                        choose the report mode
  -c CONFIG, --config CONFIG
                        check the security hardening options in the kernel Kconfig file (also
                        supports *.gz files)
  -l CMDLINE, --cmdline CMDLINE
                        check the security hardening options in the kernel cmdline file
  -p {X86_64,X86_32,ARM64,ARM}, --print {X86_64,X86_32,ARM64,ARM}
                        print the security hardening recommendations for the selected
                        microarchitecture
  -g {X86_64,X86_32,ARM64,ARM}, --generate {X86_64,X86_32,ARM64,ARM}
                        generate a Kconfig fragment with the security hardening options for the
                        selected microarchitecture
$
```



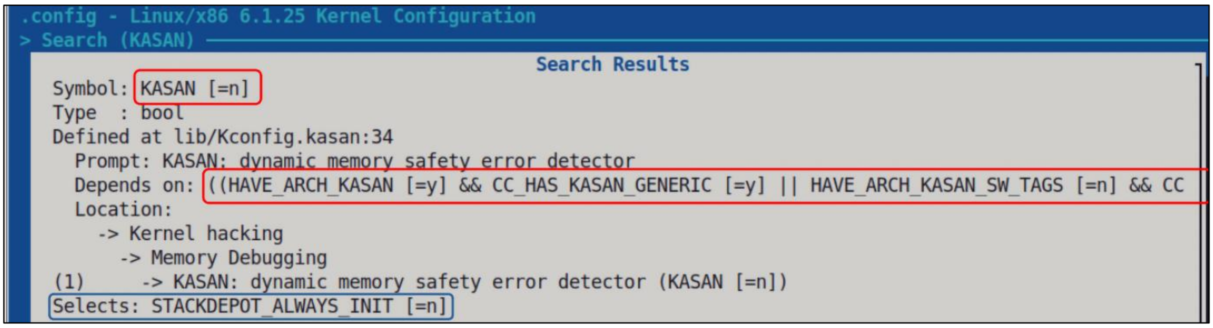
```
.config - Linux/x86 6.1.25 Kernel Configuration
> Device Drivers > Network device support > Ethernet driver support ────────
                                    Ethernet driver support
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus
  letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes featu
  to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M>
  capable
 ┌──────────────^(-)──────────────────────────────────────────────────────────────
 │              < >   Dave ethernet support (DNET)
 │              [*]   Digital Equipment devices
 │              [*]     DEC - Tulip devices
 │              < >       Early DECchip Tulip (dc2104x) PCI support
 │              < >       DECchip Tulip (dc2114x) PCI support
 │              < >       Winbond W89c840 Ethernet support
 │              < >       Davicom DM910x/DM980x support
 │              < >       ULi M526x controller support
 │              [*]   D-Link devices
 │              < >     DL2000/TC902x/IP1000A-based Gigabit Ethernet support
 │              < >     Sundance Alta support
 │              [*]   Emulex devices
 │              < >     ServerEngines' 10Gbps NIC - BladeEngine
 │              [*]   Engleder devices
 │              < >     TSN endpoint support
 │              [*]   EZchip devices
 │              [*]   Fungible devices
 │              < >     Fungible Ethernet device driver
 │              [*]   Google Devices
 │              < >     Google Virtual NIC (gVNIC) support
```



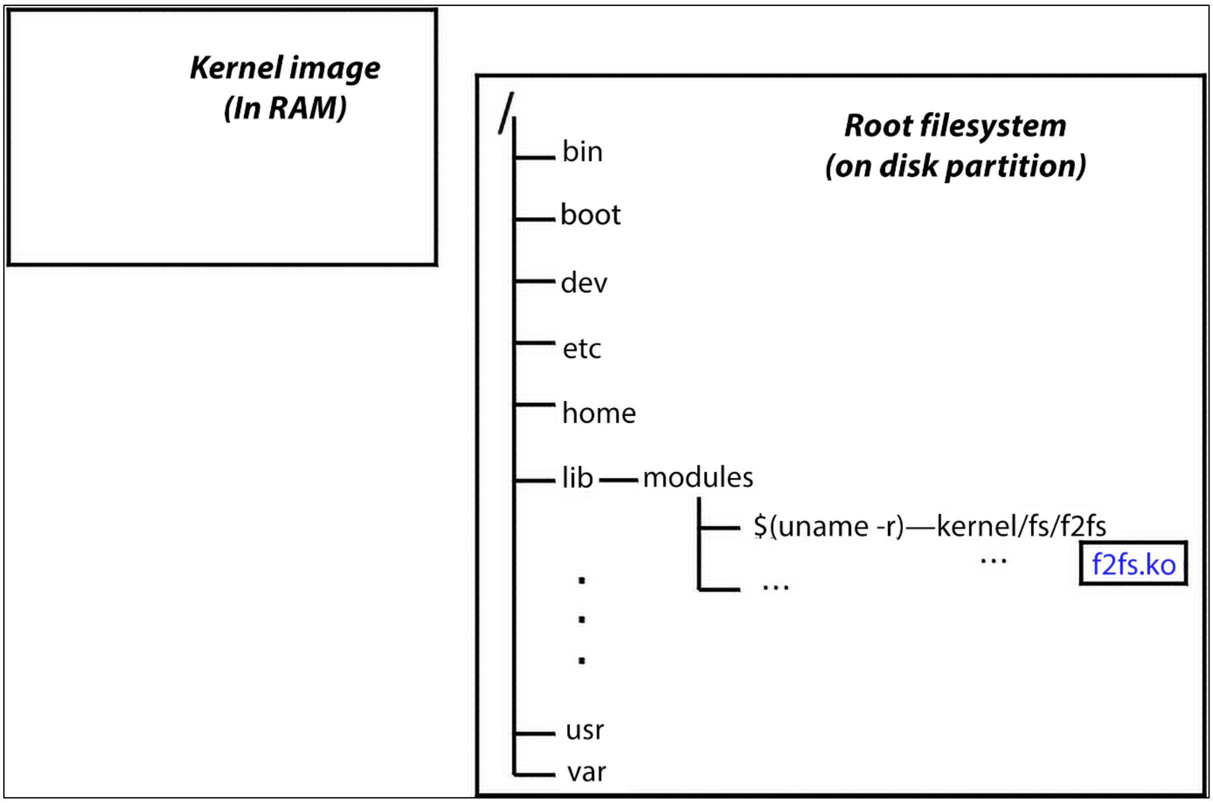
```
          which is done within the script "scripts/setlocalversion".)

config LKP_OPTION1
        bool "Test case for LKP 2e book/Ch 2: creating a new menu item in kernel config"
        default n
        help
          This option is merely a dummy or 'test' one; it's simply to have readers
          of this book - 'Linux Kernel Programming', 2nd Ed, Kaiwan NB, Packt -
          try out the creation of a few menu items within the kernel config.

          Within the 'make menuconfig', you can experiment: set this option to
          'y' (on), save and exit, and see the effect this has by doing:
          grep "CONFIG_LKP_OPTION1" .config

          If unsure, say N

config BUILD_SALT
        string "Build ID Salt"
```

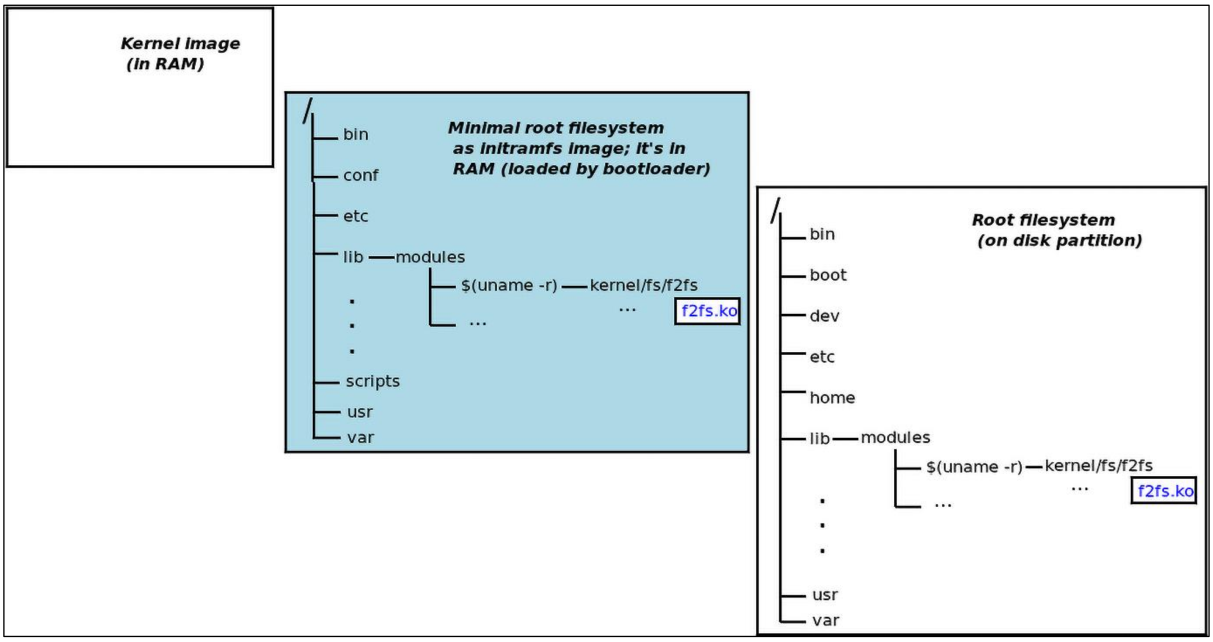


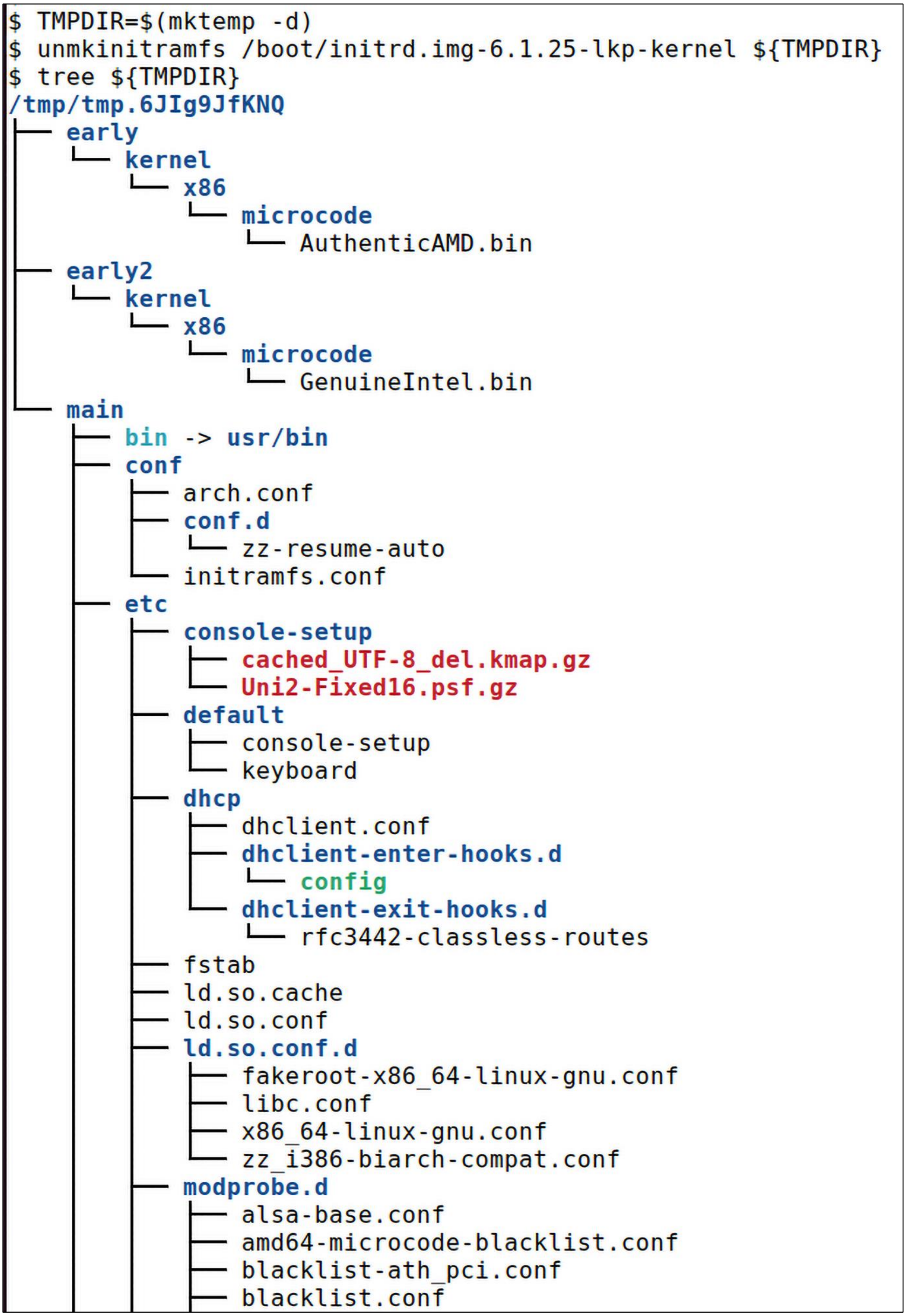
```
.config - Linux/x86 6.1.25 Kernel Configuration
> General setup
                              General setup
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).  Highlighted
  letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc>
  to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module
  capable

                 [ ] Compile also drivers which will not load
                 [ ] Compile the kernel with warnings as errors
                 (-lkp-kernel) Local version - append to kernel release
                 [ ] Automatically append version information to the version string
                 [ ] Test case for LKP 2e book/Ch 2: creating a new menu item in kernel config (NEW)
                 ()  Build ID Salt
                     Kernel compression mode (ZSTD)  --->
```



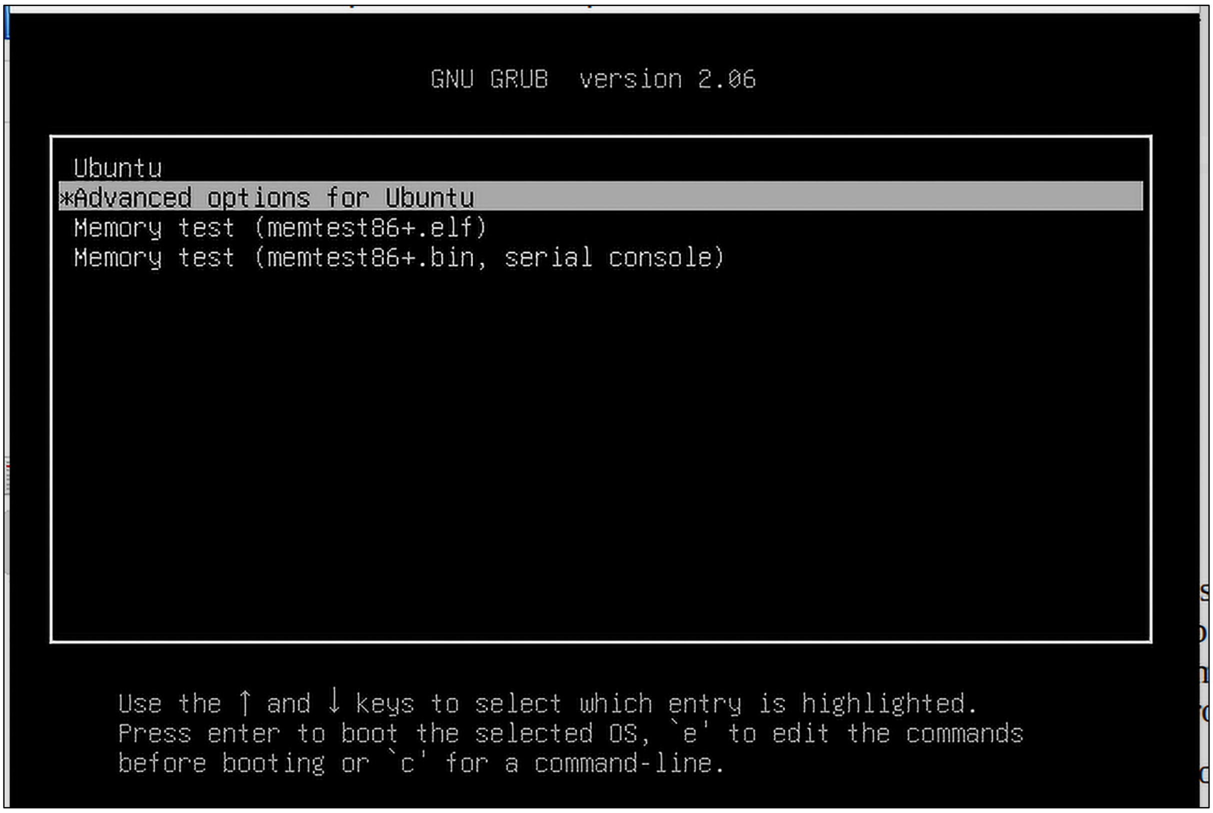
```
.config - Linux/x86 6.1.25 Kernel Configuration
> Search (KASAN)
                              Search Results
Symbol: KASAN [=n]
Type  : bool
Defined at lib/Kconfig.kasan:34
  Prompt: KASAN: dynamic memory safety error detector
  Depends on: ((HAVE_ARCH_KASAN [=y] && CC_HAS_KASAN_GENERIC [=y] || HAVE_ARCH_KASAN_SW_TAGS [=n] && CC
  Location:
    -> Kernel hacking
      -> Memory Debugging
(1)     -> KASAN: dynamic memory safety error detector (KASAN [=n])
Selects: STACKDEPOT_ALWAYS_INIT [=n]
```


Kernel image
(In RAM)
Root filesystem
(on disk partition)
/
bin
boot
dev
etc
home
lib
modules
$(uname -r)—kernel/fs/f2fs
...
...
f2fs.ko
usr
var

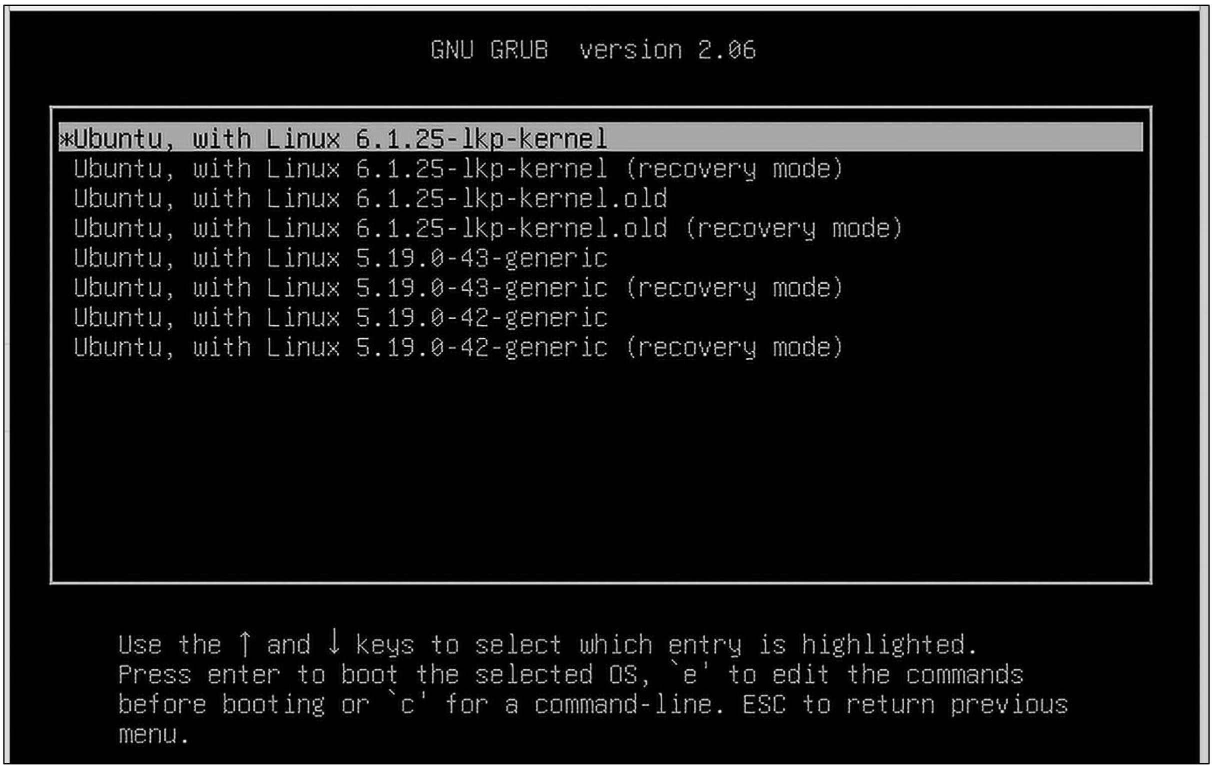

Kernel image
(In RAM)
Minimal root filesystem
as initramfs image; it's in
RAM (loaded by bootloader)
/
bin
conf
etc
lib
modules
$(uname -r)
kernel/fs/f2fs
...
...
f2fs.ko
scripts
usr
var
Root filesystem
(on disk partition)
/
bin
boot
dev
etc
home
lib
modules
$(uname -r)
kernel/fs/f2fs
...
...
f2fs.ko
usr
var



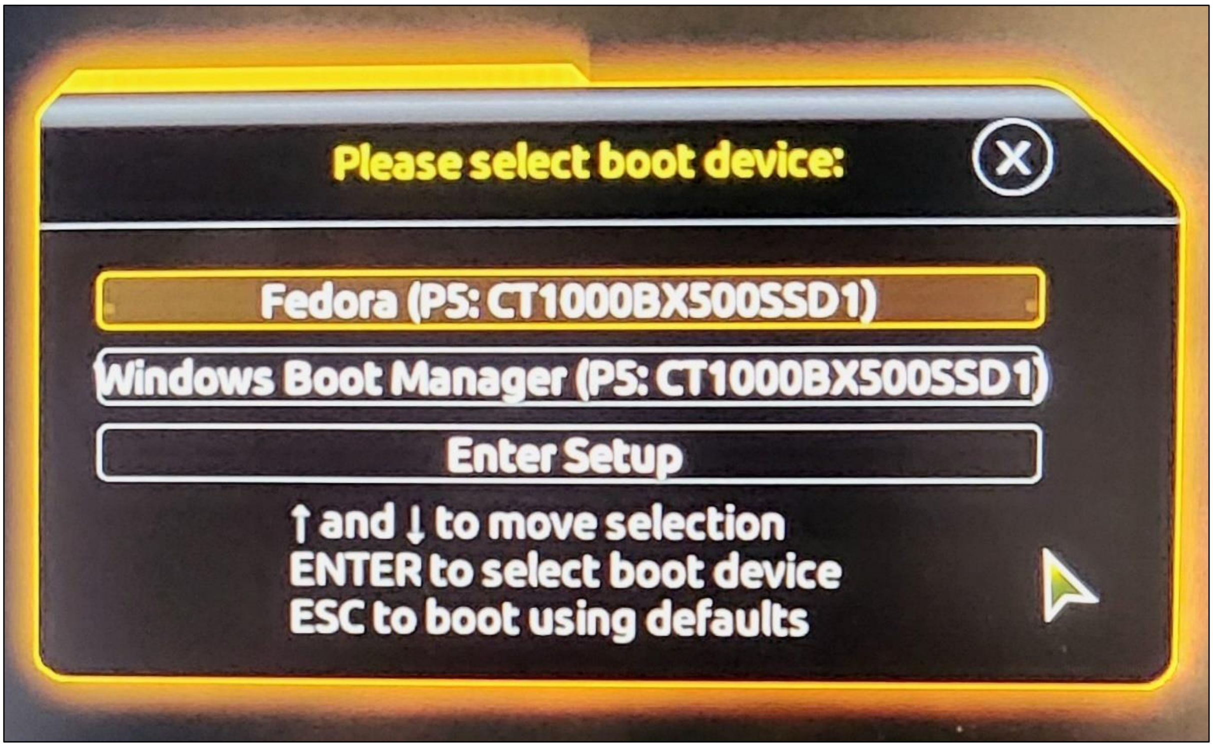
```
$ TMPDIR=$(mktemp -d)
$ unmkinitramfs /boot/initrd.img-6.1.25-lkp-kernel ${TMPDIR}
$ tree ${TMPDIR}
/tmp/tmp.6JIg9JfKNQ
├── early
│   └── kernel
│       └── x86
│           └── microcode
│               └── AuthenticAMD.bin
├── early2
│   └── kernel
│       └── x86
│           └── microcode
│               └── GenuineIntel.bin
└── main
    ├── bin -> usr/bin
    ├── conf
    │   ├── arch.conf
    │   ├── conf.d
    │   │   └── zz-resume-auto
    │   └── initramfs.conf
    ├── etc
    │   ├── console-setup
    │   │   ├── cached_UTF-8_del.kmap.gz
    │   │   └── Uni2-Fixed16.psf.gz
    │   ├── default
    │   │   ├── console-setup
    │   │   └── keyboard
    │   ├── dhcp
    │   │   ├── dhclient.conf
    │   │   ├── dhclient-enter-hooks.d
    │   │   │   └── config
    │   │   └── dhclient-exit-hooks.d
    │   │       └── rfc3442-classless-routes
    │   ├── fstab
    │   ├── ld.so.cache
    │   ├── ld.so.conf
    │   ├── ld.so.conf.d
    │   │   ├── fakeroot-x86_64-linux-gnu.conf
    │   │   ├── libc.conf
    │   │   ├── x86_64-linux-gnu.conf
    │   │   └── zz_i386-biarch-compat.conf
    │   ├── modprobe.d
    │   │   ├── alsa-base.conf
    │   │   ├── amd64-microcode-blacklist.conf
    │   │   ├── blacklist-ath_pci.conf
    │   │   ├── blacklist.conf
```



```
                      GNU GRUB  version 2.06

 Ubuntu
*Advanced options for Ubuntu
 Memory test (memtest86+.elf)
 Memory test (memtest86+.bin, serial console)

    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.
```



```
                      GNU GRUB  version 2.06

*Ubuntu, with Linux 6.1.25-lkp-kernel
 Ubuntu, with Linux 6.1.25-lkp-kernel (recovery mode)
 Ubuntu, with Linux 6.1.25-lkp-kernel.old
 Ubuntu, with Linux 6.1.25-lkp-kernel.old (recovery mode)
 Ubuntu, with Linux 5.19.0-43-generic
 Ubuntu, with Linux 5.19.0-43-generic (recovery mode)
 Ubuntu, with Linux 5.19.0-42-generic
 Ubuntu, with Linux 5.19.0-42-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.
```



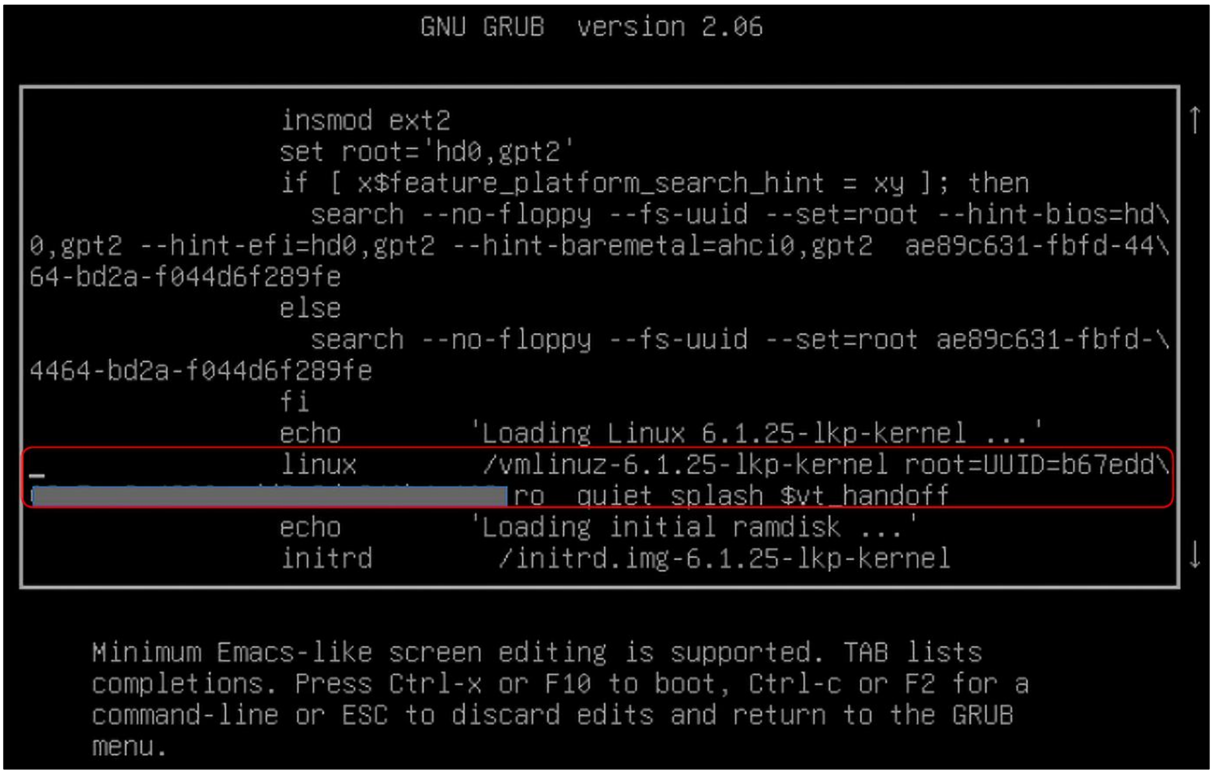
Please select boot device:

Fedora (P5: CT1000BX500SSD1)

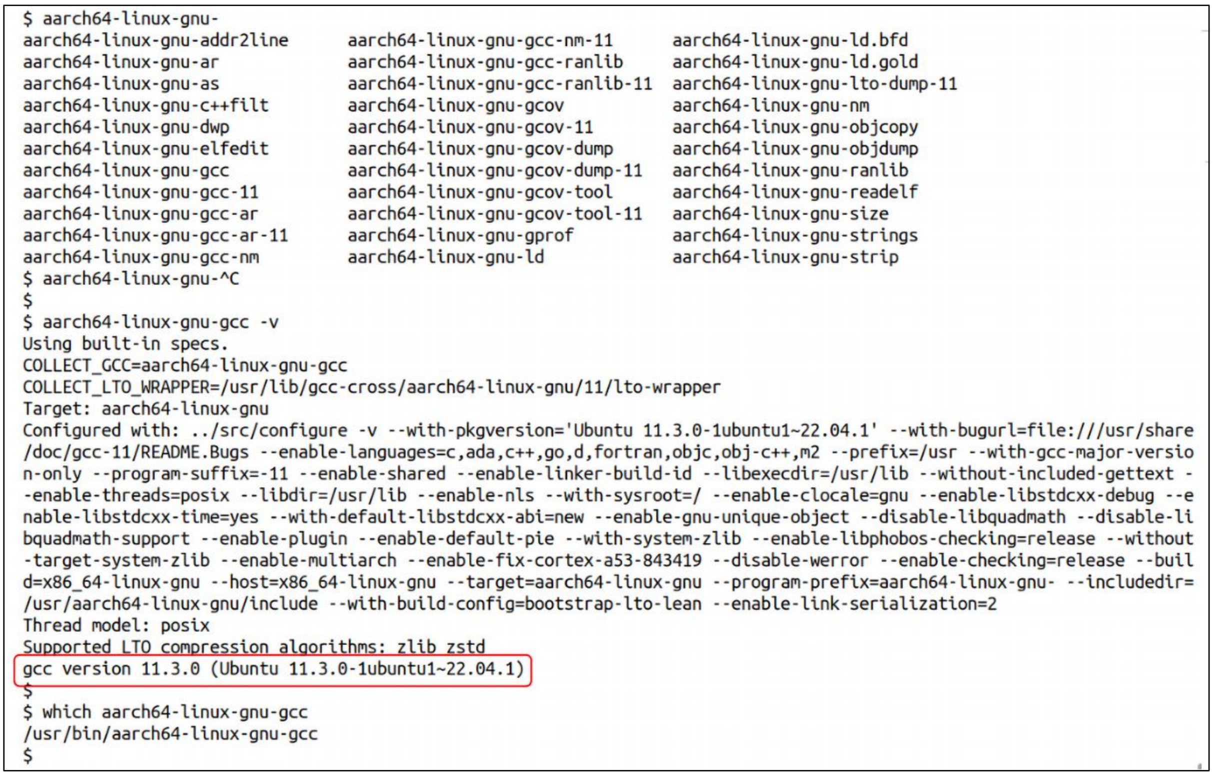
Windows Boot Manager (P5: CT1000BX500SSD1)

Enter Setup

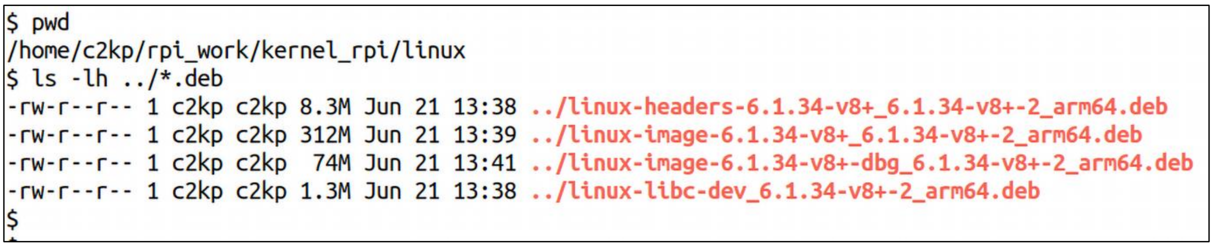
↑ and ↓ to move selection
ENTER to select boot device
ESC to boot using defaults



```
                         GNU GRUB  version 2.06

                 insmod ext2
                 set root='hd0,gpt2'
                 if [ x$feature_platform_search_hint = xy ]; then
                   search --no-floppy --fs-uuid --set=root --hint-bios=hd\
0,gpt2 --hint-efi=hd0,gpt2 --hint-baremetal=ahci0,gpt2  ae89c631-fbfd-44\
64-bd2a-f044d6f289fe
                 else
                   search --no-floppy --fs-uuid --set=root ae89c631-fbfd-\
4464-bd2a-f044d6f289fe
                 fi
                 echo         'Loading Linux 6.1.25-lkp-kernel ...'
_                linux         /vmlinuz-6.1.25-lkp-kernel root=UUID=b67edd\
[redacted] ro  quiet splash $vt_handoff
                 echo         'Loading initial ramdisk ...'
                 initrd        /initrd.img-6.1.25-lkp-kernel

    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.
```

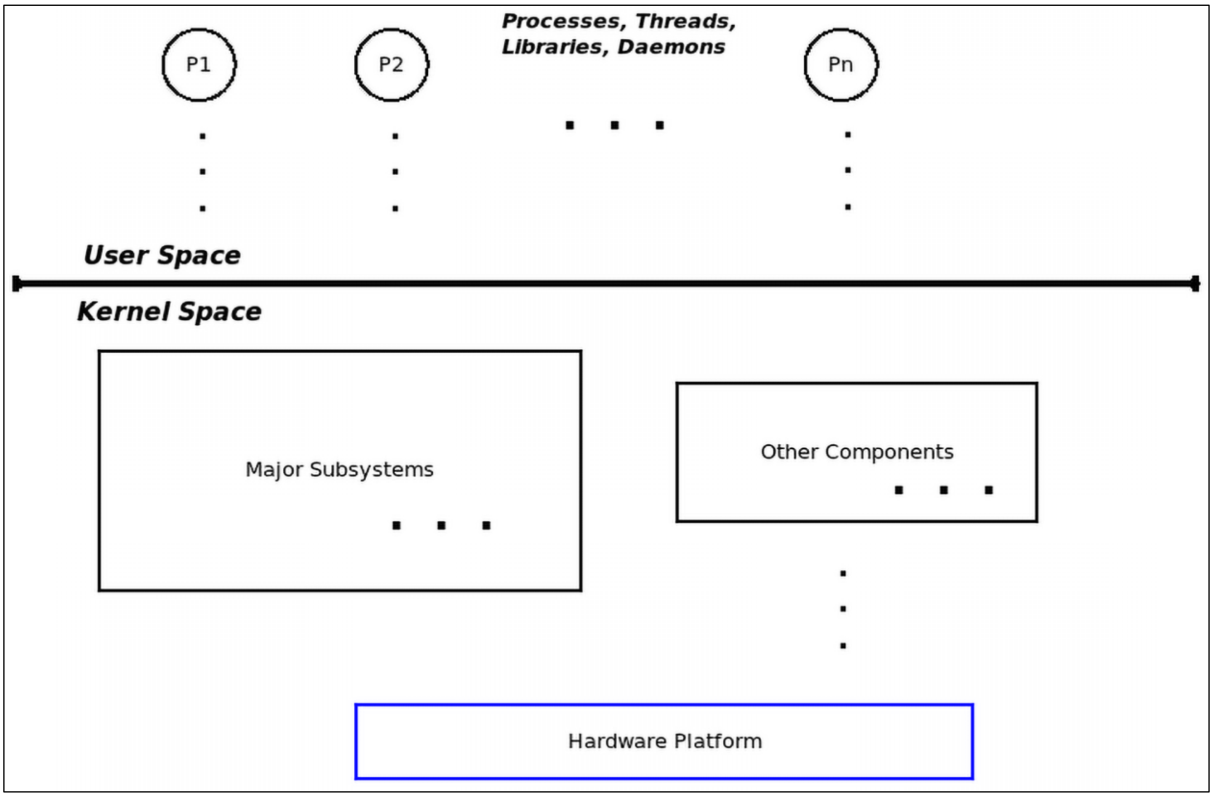


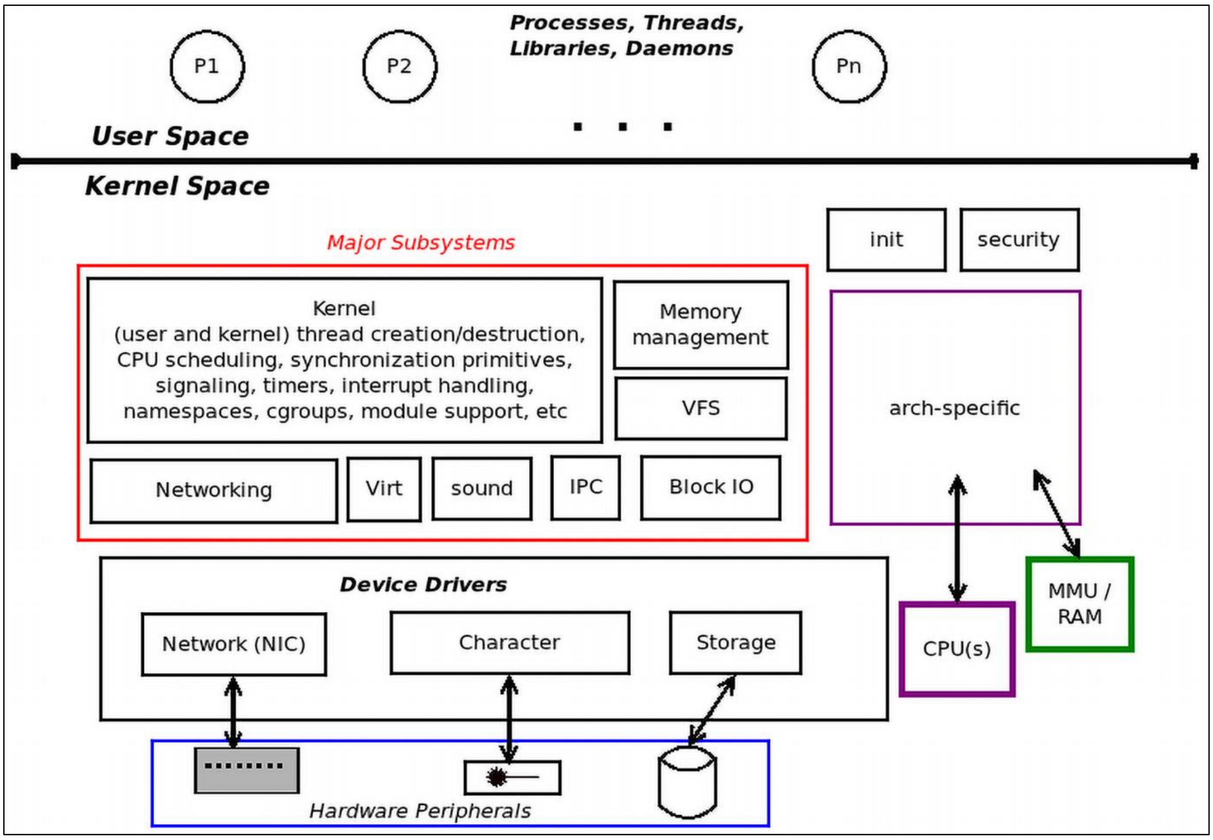
```
$ aarch64-linux-gnu-
aarch64-linux-gnu-addr2line      aarch64-linux-gnu-gcc-nm-11      aarch64-linux-gnu-ld.bfd
aarch64-linux-gnu-ar             aarch64-linux-gnu-gcc-ranlib     aarch64-linux-gnu-ld.gold
aarch64-linux-gnu-as             aarch64-linux-gnu-gcc-ranlib-11  aarch64-linux-gnu-lto-dump-11
aarch64-linux-gnu-c++filt        aarch64-linux-gnu-gcov           aarch64-linux-gnu-nm
aarch64-linux-gnu-dwp            aarch64-linux-gnu-gcov-11        aarch64-linux-gnu-objcopy
aarch64-linux-gnu-elfedit        aarch64-linux-gnu-gcov-dump      aarch64-linux-gnu-objdump
aarch64-linux-gnu-gcc            aarch64-linux-gnu-gcov-dump-11   aarch64-linux-gnu-ranlib
aarch64-linux-gnu-gcc-11         aarch64-linux-gnu-gcov-tool      aarch64-linux-gnu-readelf
aarch64-linux-gnu-gcc-ar         aarch64-linux-gnu-gcov-tool-11   aarch64-linux-gnu-size
aarch64-linux-gnu-gcc-ar-11      aarch64-linux-gnu-gprof          aarch64-linux-gnu-strings
aarch64-linux-gnu-gcc-nm         aarch64-linux-gnu-ld             aarch64-linux-gnu-strip
$ aarch64-linux-gnu-^C
$
$ aarch64-linux-gnu-gcc -v
Using built-in specs.
COLLECT_GCC=aarch64-linux-gnu-gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc-cross/aarch64-linux-gnu/11/lto-wrapper
Target: aarch64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 11.3.0-1ubuntu1~22.04.1' --with-bugurl=file:///usr/share
/doc/gcc-11/README.Bugs --enable-languages=c,ada,c++,go,d,fortran,objc,obj-c++,m2 --prefix=/usr --with-gcc-major-versio
n-only --program-suffix=-11 --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext -
-enable-threads=posix --libdir=/usr/lib --enable-nls --with-sysroot=/ --enable-clocale=gnu --enable-libstdcxx-debug --e
nable-libstdcxx-time=yes --with-default-libstdcxx-abi=new --enable-gnu-unique-object --disable-libquadmath --disable-li
bquadmath-support --enable-plugin --enable-default-pie --with-system-zlib --enable-libphobos-checking=release --without
-target-system-zlib --enable-multiarch --enable-fix-cortex-a53-843419 --disable-werror --enable-checking=release --buil
d=x86_64-linux-gnu --host=x86_64-linux-gnu --target=aarch64-linux-gnu --program-prefix=aarch64-linux-gnu- --includedir=
/usr/aarch64-linux-gnu/include --with-build-config=bootstrap-lto-lean --enable-link-serialization=2
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 11.3.0 (Ubuntu 11.3.0-1ubuntu1~22.04.1)
$
$ which aarch64-linux-gnu-gcc
/usr/bin/aarch64-linux-gnu-gcc
$
```

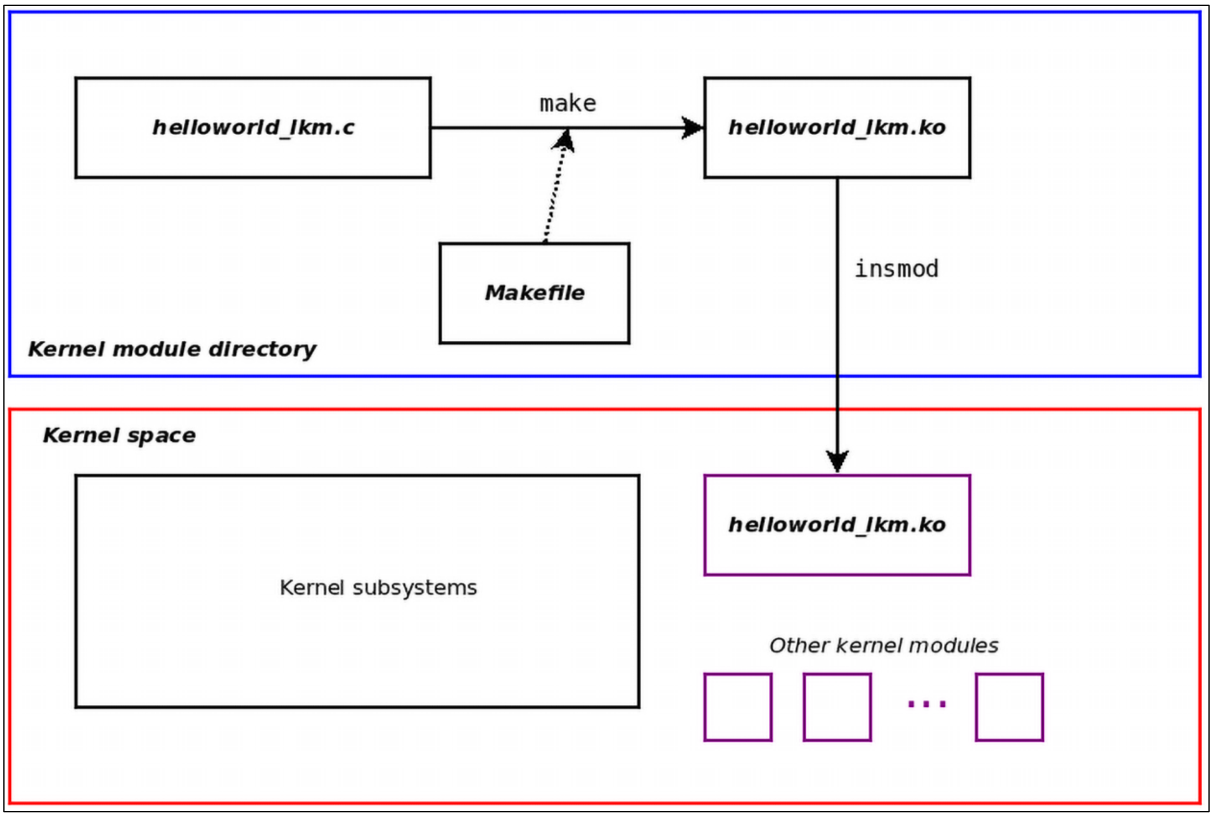


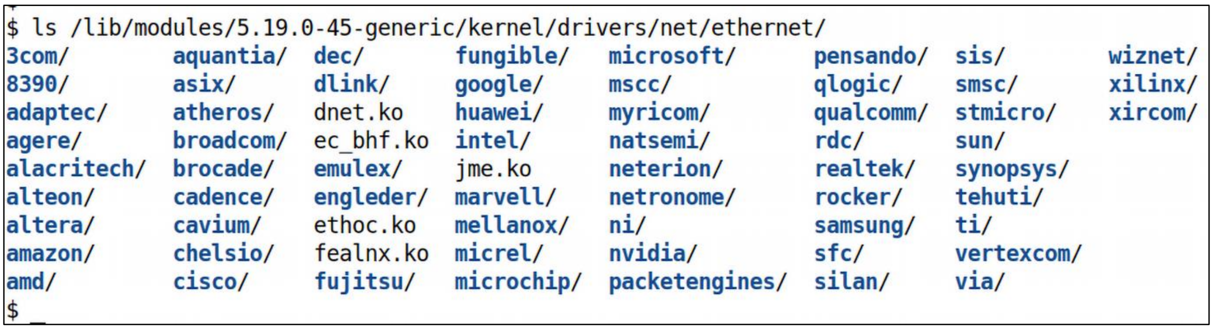
```
$ pwd
/home/c2kp/rpi_work/kernel_rpi/linux
$ ls -lh ../*.deb
-rw-r--r-- 1 c2kp c2kp 8.3M Jun 21 13:38 ../linux-headers-6.1.34-v8+_6.1.34-v8+-2_arm64.deb
-rw-r--r-- 1 c2kp c2kp 312M Jun 21 13:39 ../linux-image-6.1.34-v8+_6.1.34-v8+-2_arm64.deb
-rw-r--r-- 1 c2kp c2kp  74M Jun 21 13:41 ../linux-image-6.1.34-v8+-dbg_6.1.34-v8+-2_arm64.deb
-rw-r--r-- 1 c2kp c2kp 1.3M Jun 21 13:38 ../linux-libc-dev_6.1.34-v8+-2_arm64.deb
$
```

Chapter 4: Writing Your First Kernel Module – Part 1

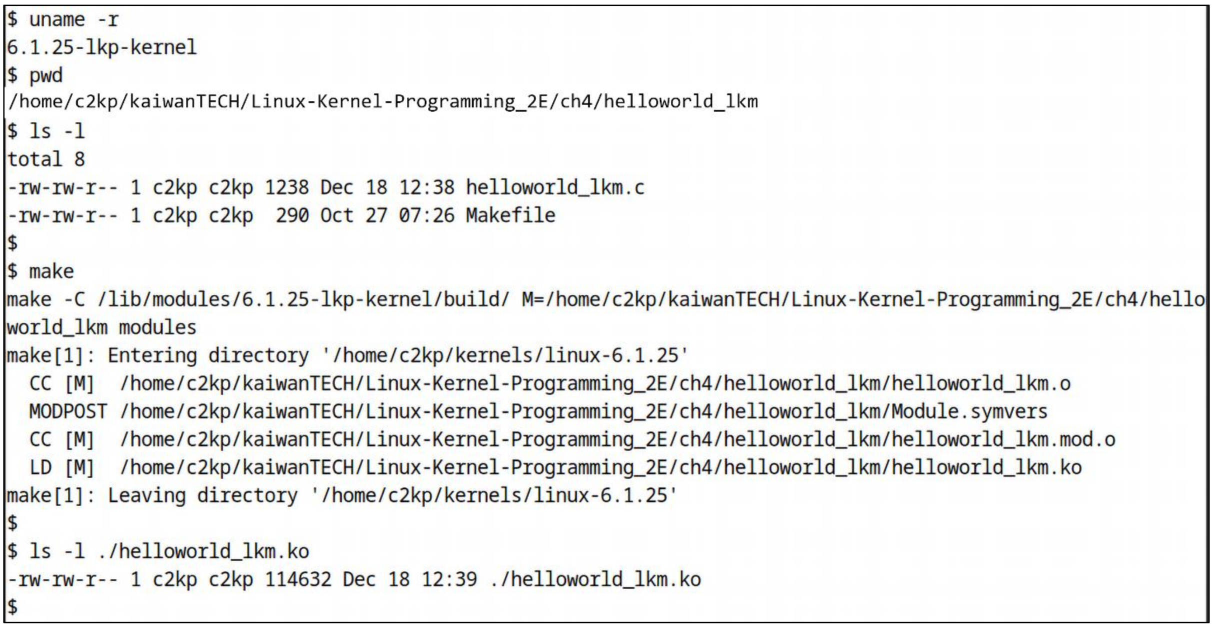




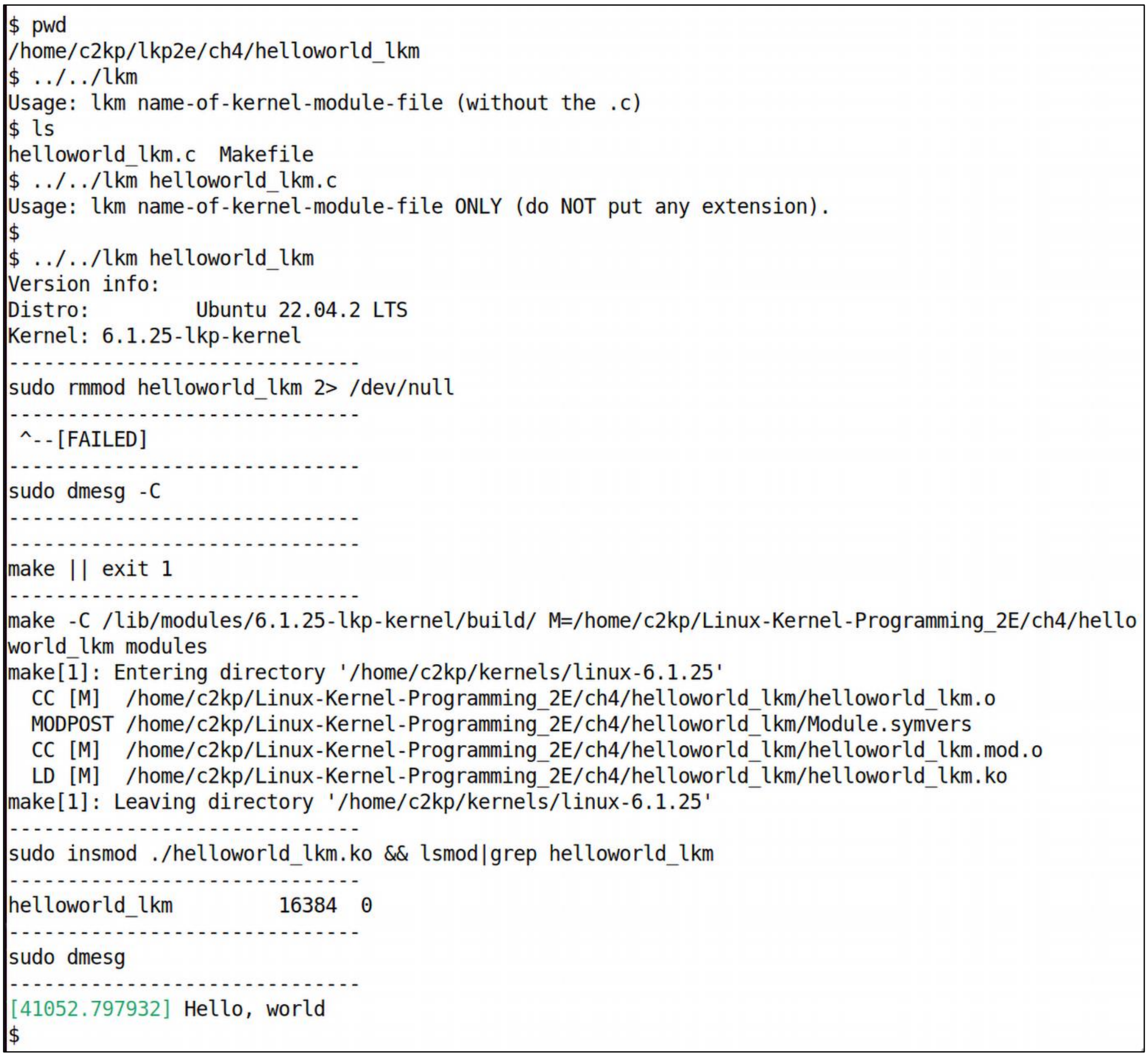




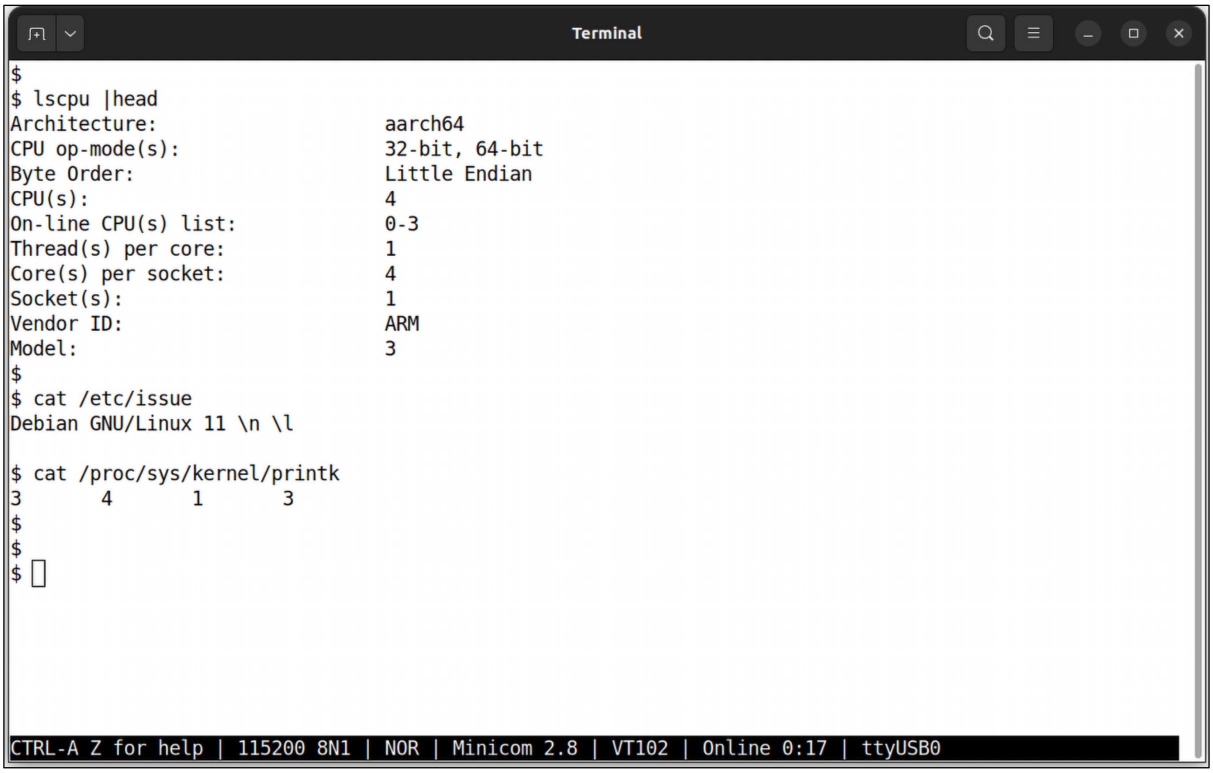
```
$ ls /lib/modules/5.19.0-45-generic/kernel/drivers/net/ethernet/
3com/       aquantia/  dec/       fungible/  microsoft/      pensando/  sis/        wiznet/
8390/       asix/      dlink/     google/    mscc/           qlogic/    smsc/       xilinx/
adaptec/    atheros/   dnet.ko    huawei/    myricom/        qualcomm/  stmicro/    xircom/
agere/      broadcom/  ec_bhf.ko  intel/     natsemi/        rdc/       sun/
alacritech/ brocade/   emulex/    jme.ko     neterion/       realtek/   synopsys/
alteon/     cadence/   engleder/  marvell/   netronome/      rocker/    tehuti/
altera/     cavium/    ethoc.ko   mellanox/  ni/             samsung/   ti/
amazon/     chelsio/   fealnx.ko  micrel/    nvidia/         sfc/       vertexcom/
amd/        cisco/     fujitsu/   microchip/ packetengines/  silan/     via/
$ _
```



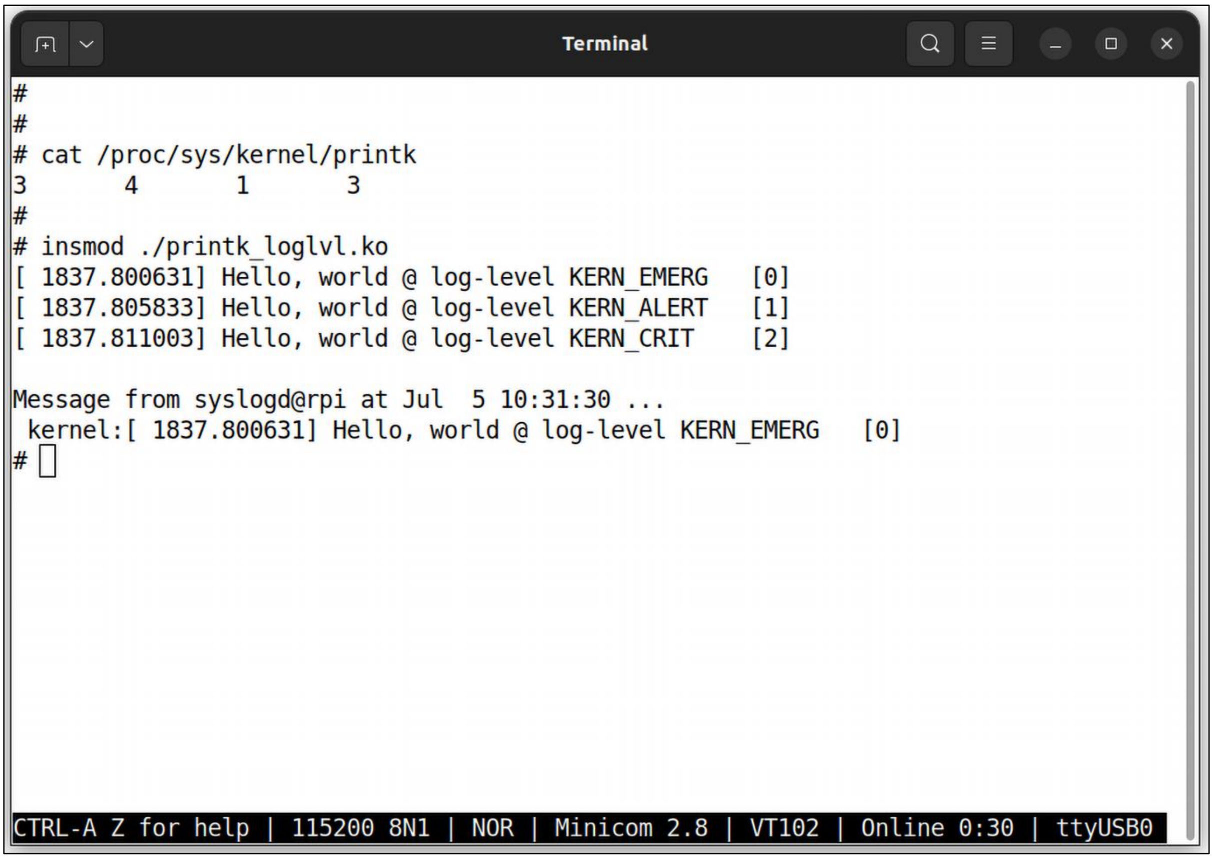
```
$ uname -r
6.1.25-lkp-kernel
$ pwd
/home/c2kp/kaiwanTECH/Linux-Kernel-Programming_2E/ch4/helloworld_lkm
$ ls -l
total 8
-rw-rw-r-- 1 c2kp c2kp 1238 Dec 18 12:38 helloworld_lkm.c
-rw-rw-r-- 1 c2kp c2kp  290 Oct 27 07:26 Makefile
$
$ make
make -C /lib/modules/6.1.25-lkp-kernel/build/ M=/home/c2kp/kaiwanTECH/Linux-Kernel-Programming_2E/ch4/hello
world_lkm modules
make[1]: Entering directory '/home/c2kp/kernels/linux-6.1.25'
  CC [M]  /home/c2kp/kaiwanTECH/Linux-Kernel-Programming_2E/ch4/helloworld_lkm/helloworld_lkm.o
  MODPOST /home/c2kp/kaiwanTECH/Linux-Kernel-Programming_2E/ch4/helloworld_lkm/Module.symvers
  CC [M]  /home/c2kp/kaiwanTECH/Linux-Kernel-Programming_2E/ch4/helloworld_lkm/helloworld_lkm.mod.o
  LD [M]  /home/c2kp/kaiwanTECH/Linux-Kernel-Programming_2E/ch4/helloworld_lkm/helloworld_lkm.ko
make[1]: Leaving directory '/home/c2kp/kernels/linux-6.1.25'
$
$ ls -l ./helloworld_lkm.ko
-rw-rw-r-- 1 c2kp c2kp 114632 Dec 18 12:39 ./helloworld_lkm.ko
$
```



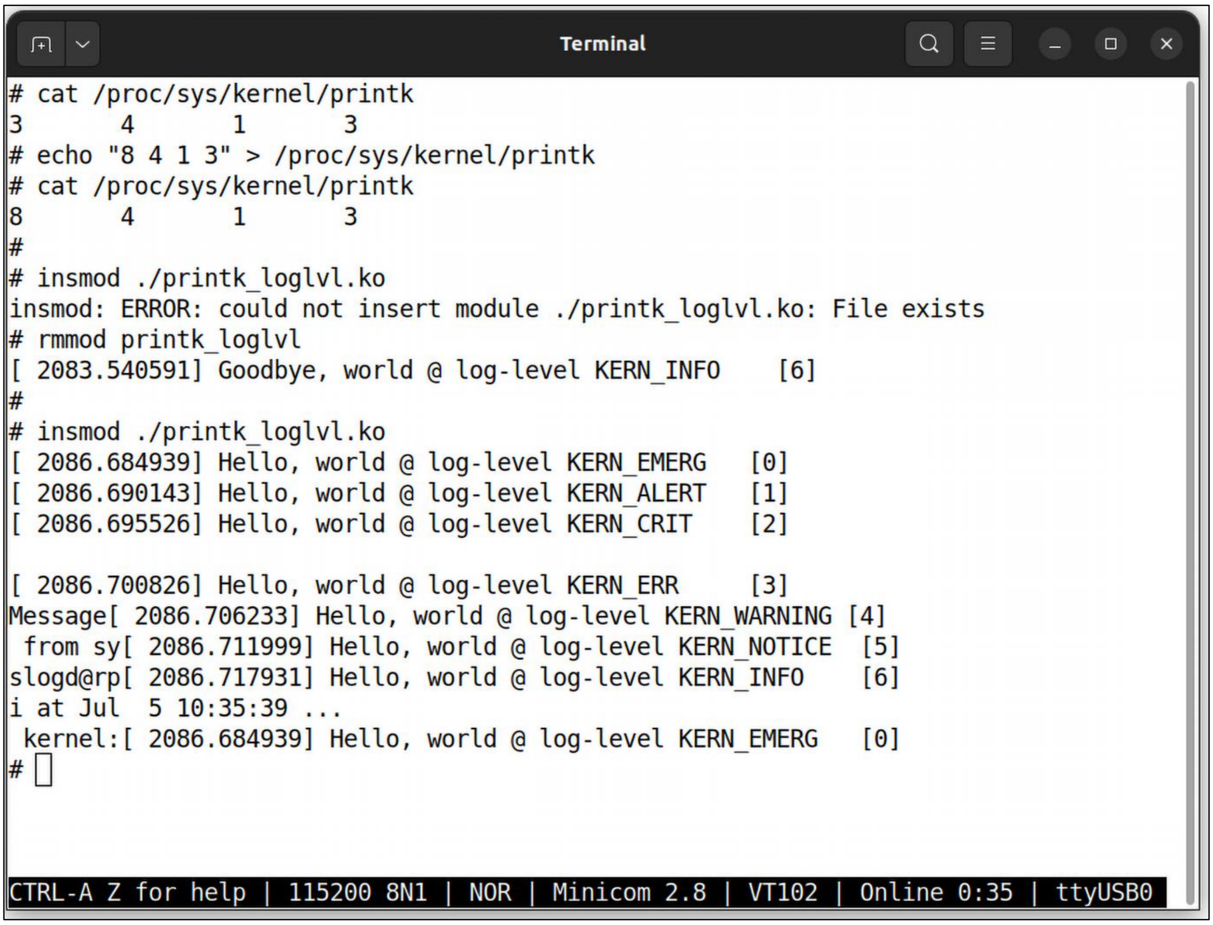
```
$ pwd
/home/c2kp/lkp2e/ch4/helloworld_lkm
$ ../../lkm
Usage: lkm name-of-kernel-module-file (without the .c)
$ ls
helloworld_lkm.c  Makefile
$ ../../lkm helloworld_lkm.c
Usage: lkm name-of-kernel-module-file ONLY (do NOT put any extension).
$
$ ../../lkm helloworld_lkm
Version info:
Distro:         Ubuntu 22.04.2 LTS
Kernel: 6.1.25-lkp-kernel
------------------------------
sudo rmmod helloworld_lkm 2> /dev/null
------------------------------
 ^--[FAILED]
------------------------------
sudo dmesg -C
------------------------------
------------------------------
make || exit 1
------------------------------
make -C /lib/modules/6.1.25-lkp-kernel/build/ M=/home/c2kp/Linux-Kernel-Programming_2E/ch4/hello
world_lkm modules
make[1]: Entering directory '/home/c2kp/kernels/linux-6.1.25'
  CC [M]  /home/c2kp/Linux-Kernel-Programming_2E/ch4/helloworld_lkm/helloworld_lkm.o
  MODPOST /home/c2kp/Linux-Kernel-Programming_2E/ch4/helloworld_lkm/Module.symvers
  CC [M]  /home/c2kp/Linux-Kernel-Programming_2E/ch4/helloworld_lkm/helloworld_lkm.mod.o
  LD [M]  /home/c2kp/Linux-Kernel-Programming_2E/ch4/helloworld_lkm/helloworld_lkm.ko
make[1]: Leaving directory '/home/c2kp/kernels/linux-6.1.25'
------------------------------
sudo insmod ./helloworld_lkm.ko && lsmod|grep helloworld_lkm
------------------------------
helloworld_lkm         16384  0
------------------------------
sudo dmesg
------------------------------
[41052.797932] Hello, world
$
```



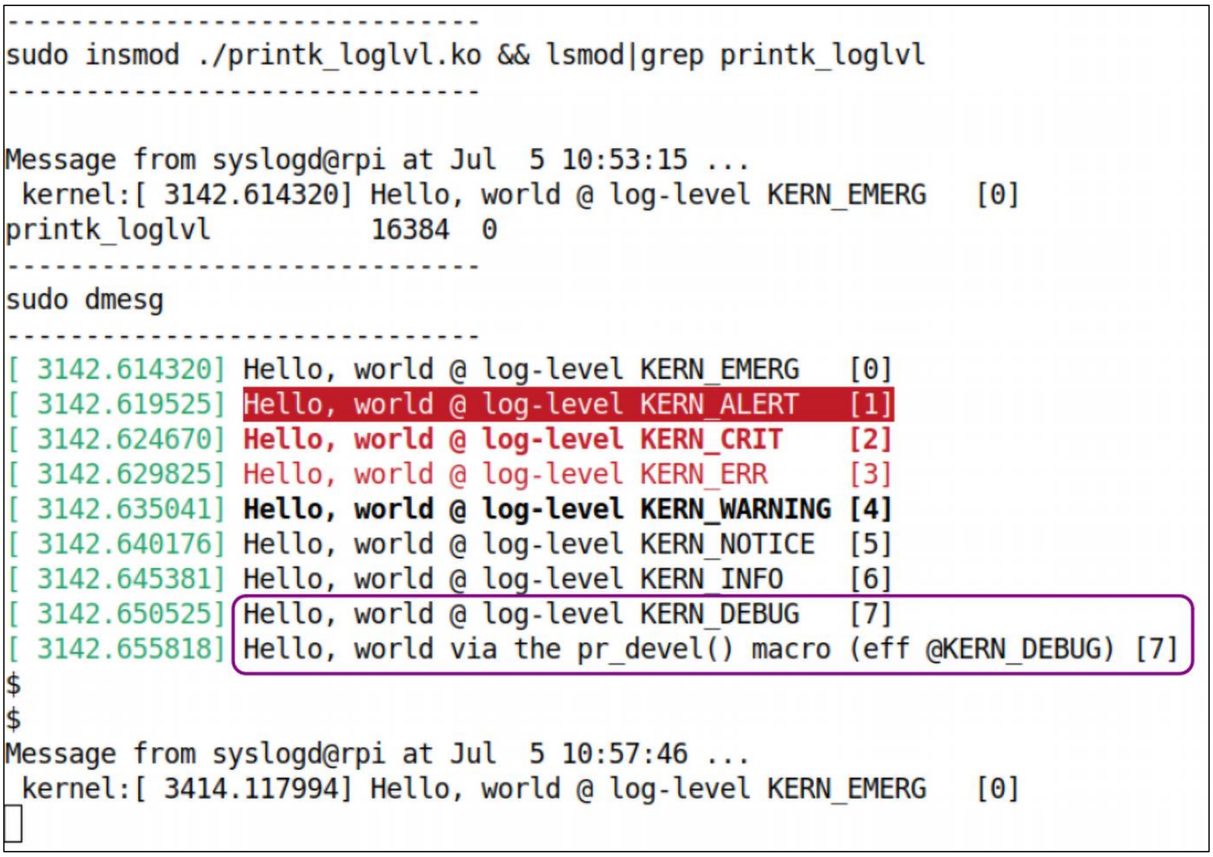
```
$
$ lscpu |head
Architecture:                    aarch64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
CPU(s):                          4
On-line CPU(s) list:             0-3
Thread(s) per core:              1
Core(s) per socket:              4
Socket(s):                       1
Vendor ID:                       ARM
Model:                           3
$
$ cat /etc/issue
Debian GNU/Linux 11 \n \l

$ cat /proc/sys/kernel/printk
3       4       1       3
$
$
$ 
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.8 | VT102 | Online 0:17 | ttyUSB0
```



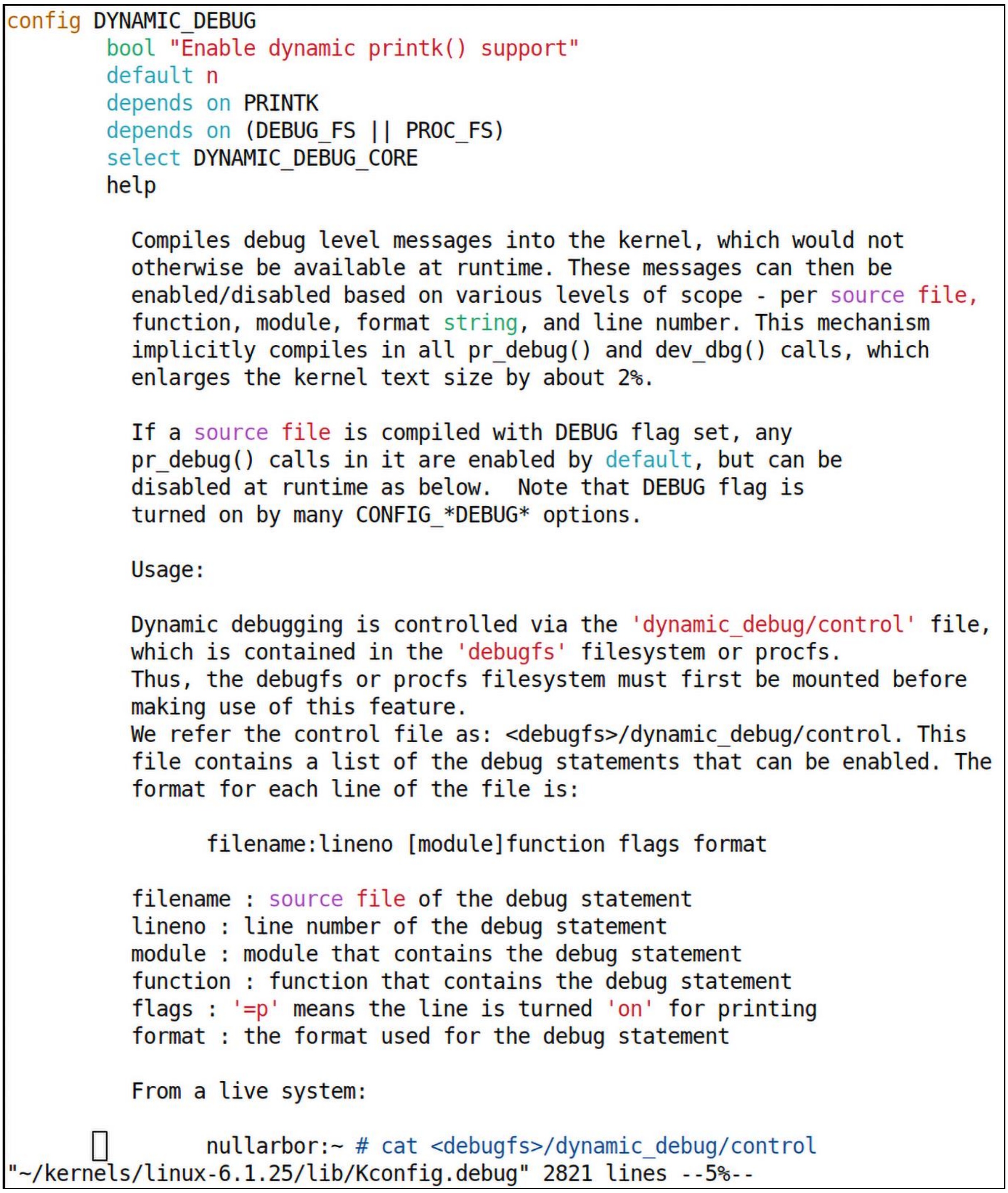
```
#
#
# cat /proc/sys/kernel/printk
3       4       1       3
#
# insmod ./printk_loglvl.ko
[ 1837.800631] Hello, world @ log-level KERN_EMERG    [0]
[ 1837.805833] Hello, world @ log-level KERN_ALERT    [1]
[ 1837.811003] Hello, world @ log-level KERN_CRIT     [2]

Message from syslogd@rpi at Jul  5 10:31:30 ...
 kernel:[ 1837.800631] Hello, world @ log-level KERN_EMERG    [0]
# 
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.8 | VT102 | Online 0:30 | ttyUSB0
```



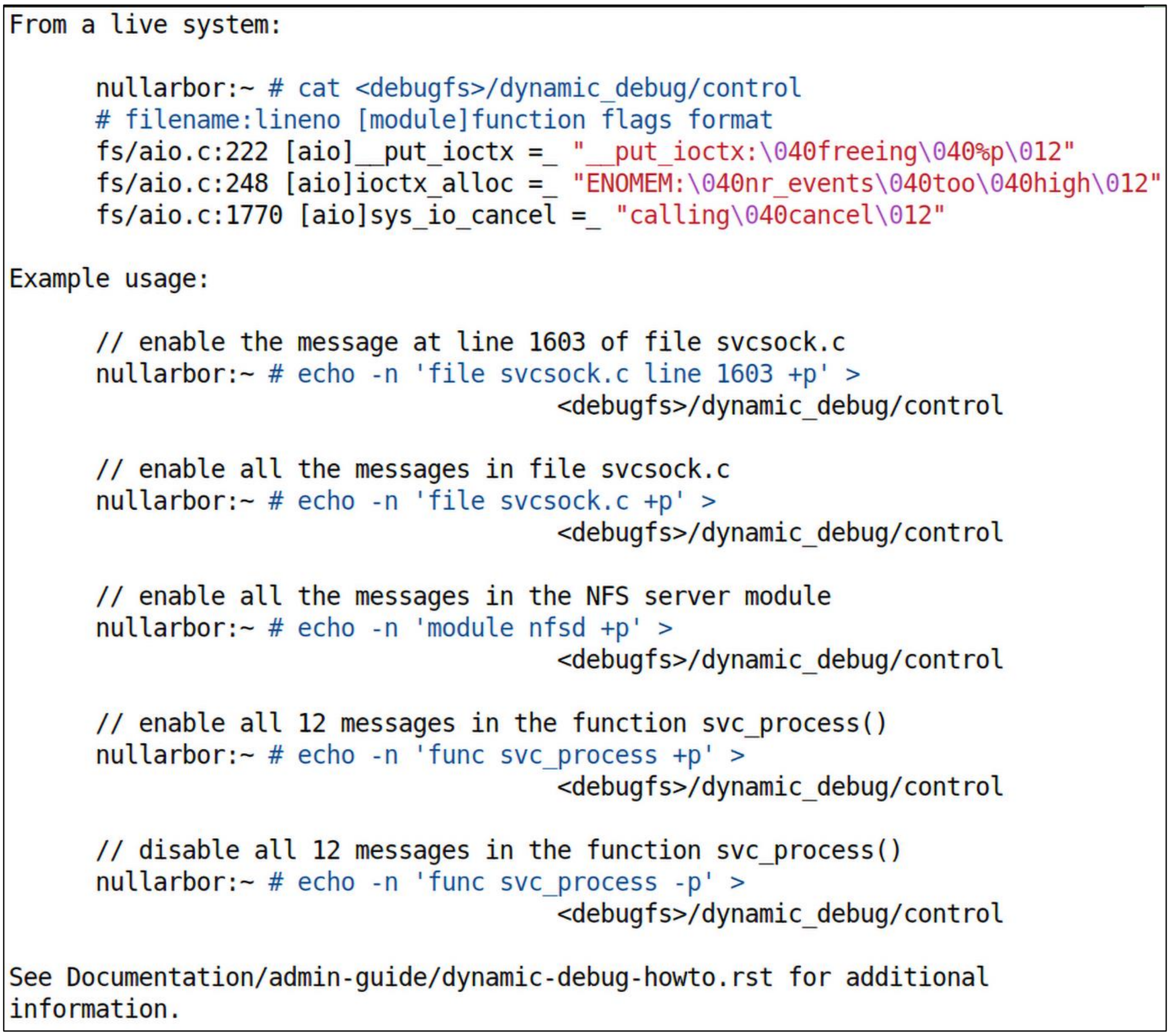
```
# cat /proc/sys/kernel/printk
3       4       1       3
# echo "8 4 1 3" > /proc/sys/kernel/printk
# cat /proc/sys/kernel/printk
8       4       1       3
#
# insmod ./printk_loglvl.ko
insmod: ERROR: could not insert module ./printk_loglvl.ko: File exists
# rmmod printk_loglvl
[ 2083.540591] Goodbye, world @ log-level KERN_INFO    [6]
#
# insmod ./printk_loglvl.ko
[ 2086.684939] Hello, world @ log-level KERN_EMERG   [0]
[ 2086.690143] Hello, world @ log-level KERN_ALERT   [1]
[ 2086.695526] Hello, world @ log-level KERN_CRIT    [2]

[ 2086.700826] Hello, world @ log-level KERN_ERR     [3]
Message[ 2086.706233] Hello, world @ log-level KERN_WARNING [4]
 from sy[ 2086.711999] Hello, world @ log-level KERN_NOTICE  [5]
slogd@rp[ 2086.717931] Hello, world @ log-level KERN_INFO    [6]
i at Jul  5 10:35:39 ...
 kernel:[ 2086.684939] Hello, world @ log-level KERN_EMERG   [0]
# 
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.8 | VT102 | Online 0:35 | ttyUSB0
```



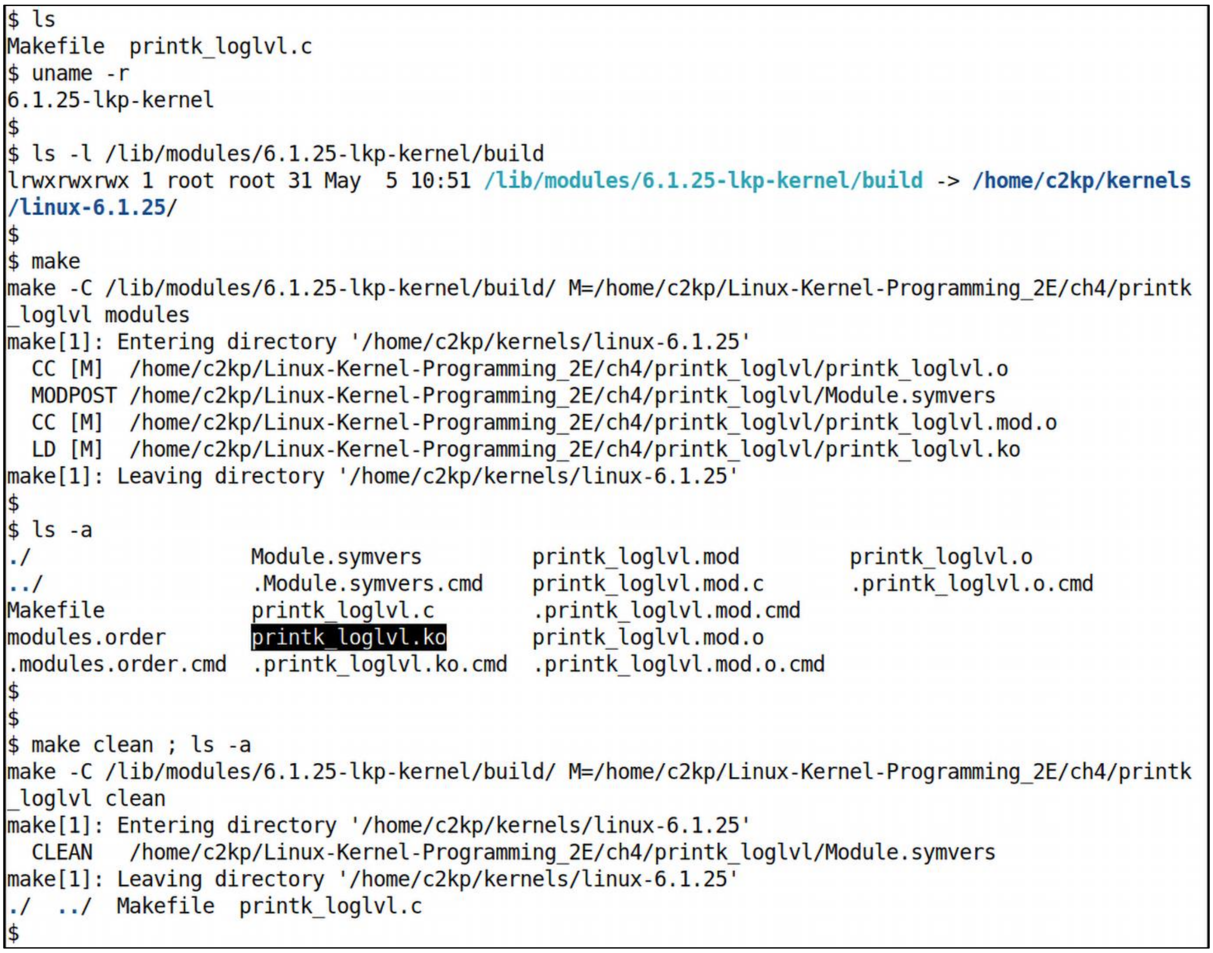
```
------------------------------
sudo insmod ./printk_loglvl.ko && lsmod|grep printk_loglvl
------------------------------

Message from syslogd@rpi at Jul  5 10:53:15 ...
 kernel:[ 3142.614320] Hello, world @ log-level KERN_EMERG   [0]
printk_loglvl          16384  0
------------------------------
sudo dmesg
------------------------------
[ 3142.614320] Hello, world @ log-level KERN_EMERG   [0]
[ 3142.619525] Hello, world @ log-level KERN_ALERT   [1]
[ 3142.624670] Hello, world @ log-level KERN_CRIT    [2]
[ 3142.629825] Hello, world @ log-level KERN_ERR     [3]
[ 3142.635041] Hello, world @ log-level KERN_WARNING [4]
[ 3142.640176] Hello, world @ log-level KERN_NOTICE  [5]
[ 3142.645381] Hello, world @ log-level KERN_INFO    [6]
[ 3142.650525] Hello, world @ log-level KERN_DEBUG   [7]
[ 3142.655818] Hello, world via the pr_devel() macro (eff @KERN_DEBUG) [7]
$
$
Message from syslogd@rpi at Jul  5 10:57:46 ...
 kernel:[ 3414.117994] Hello, world @ log-level KERN_EMERG   [0]
```



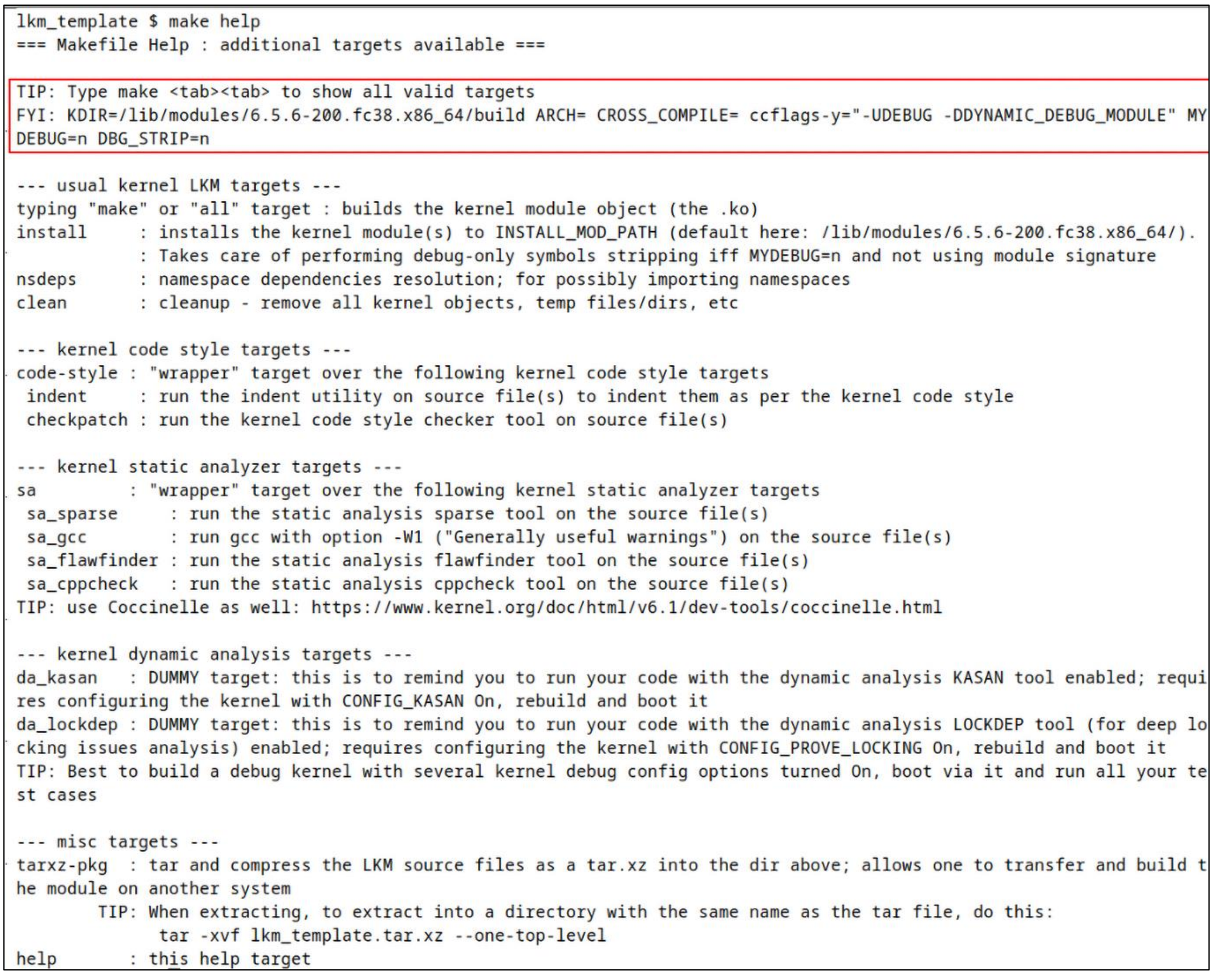
```
config DYNAMIC_DEBUG
        bool "Enable dynamic printk() support"
        default n
        depends on PRINTK
        depends on (DEBUG_FS || PROC_FS)
        select DYNAMIC_DEBUG_CORE
        help

          Compiles debug level messages into the kernel, which would not
          otherwise be available at runtime. These messages can then be
          enabled/disabled based on various levels of scope - per source file,
          function, module, format string, and line number. This mechanism
          implicitly compiles in all pr_debug() and dev_dbg() calls, which
          enlarges the kernel text size by about 2%.

          If a source file is compiled with DEBUG flag set, any
          pr_debug() calls in it are enabled by default, but can be
          disabled at runtime as below.  Note that DEBUG flag is
          turned on by many CONFIG_*DEBUG* options.

          Usage:

          Dynamic debugging is controlled via the 'dynamic_debug/control' file,
          which is contained in the 'debugfs' filesystem or procfs.
          Thus, the debugfs or procfs filesystem must first be mounted before
          making use of this feature.
          We refer the control file as: <debugfs>/dynamic_debug/control. This
          file contains a list of the debug statements that can be enabled. The
          format for each line of the file is:

                filename:lineno [module]function flags format

          filename : source file of the debug statement
          lineno : line number of the debug statement
          module : module that contains the debug statement
          function : function that contains the debug statement
          flags : '=p' means the line is turned 'on' for printing
          format : the format used for the debug statement

          From a live system:

                nullarbor:~ # cat <debugfs>/dynamic_debug/control
"~/kernels/linux-6.1.25/lib/Kconfig.debug" 2821 lines --5%--
```



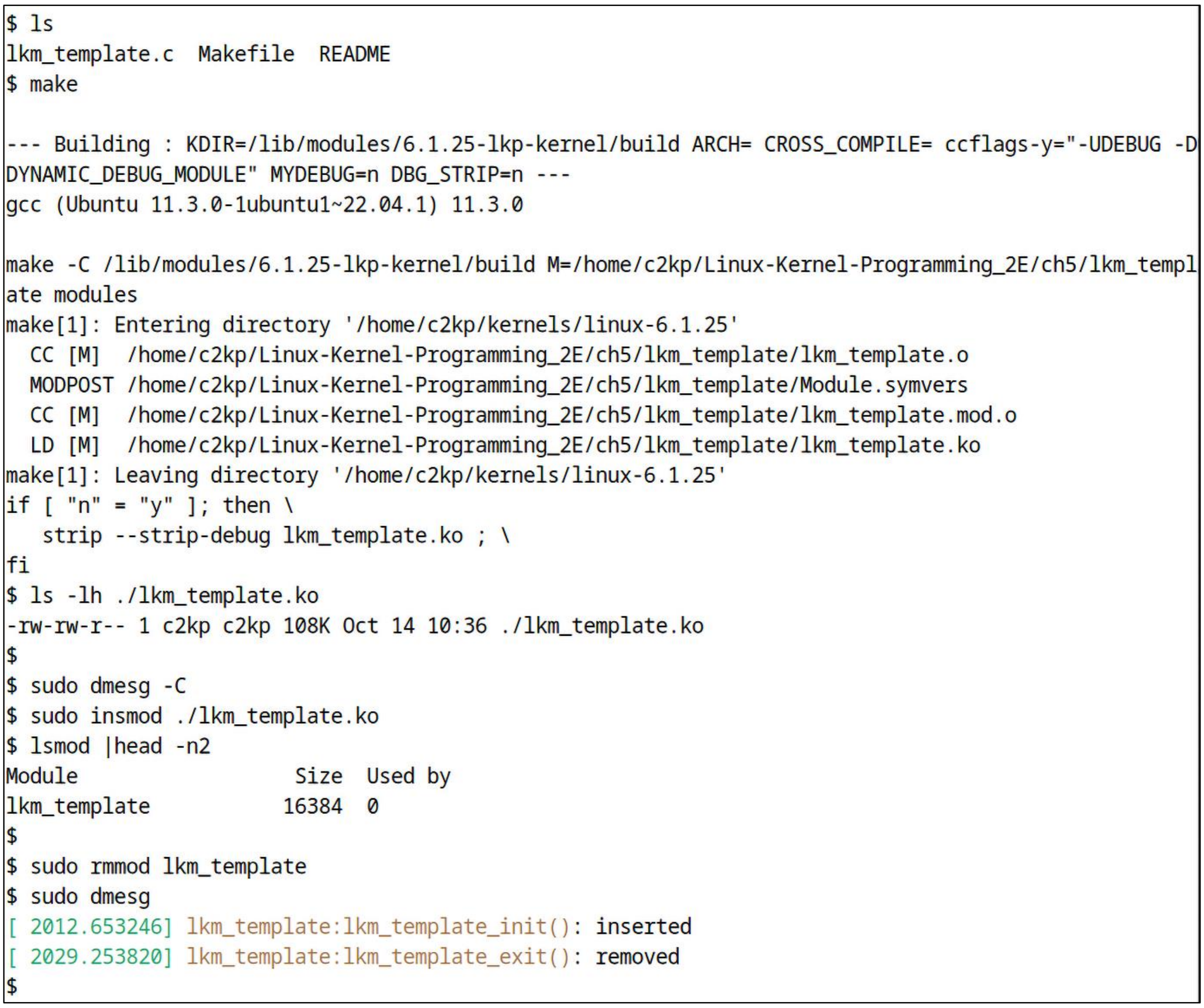
```
From a live system:

      nullarbor:~ # cat <debugfs>/dynamic_debug/control
      # filename:lineno [module]function flags format
      fs/aio.c:222 [aio]__put_ioctx =_ "__put_ioctx:\040freeing\040%p\012"
      fs/aio.c:248 [aio]ioctx_alloc =_ "ENOMEM:\040nr_events\040too\040high\012"
      fs/aio.c:1770 [aio]sys_io_cancel =_ "calling\040cancel\012"

Example usage:

      // enable the message at line 1603 of file svcsock.c
      nullarbor:~ # echo -n 'file svcsock.c line 1603 +p' >
                                      <debugfs>/dynamic_debug/control

      // enable all the messages in file svcsock.c
      nullarbor:~ # echo -n 'file svcsock.c +p' >
                                      <debugfs>/dynamic_debug/control

      // enable all the messages in the NFS server module
      nullarbor:~ # echo -n 'module nfsd +p' >
                                      <debugfs>/dynamic_debug/control

      // enable all 12 messages in the function svc_process()
      nullarbor:~ # echo -n 'func svc_process +p' >
                                      <debugfs>/dynamic_debug/control

      // disable all 12 messages in the function svc_process()
      nullarbor:~ # echo -n 'func svc_process -p' >
                                      <debugfs>/dynamic_debug/control

See Documentation/admin-guide/dynamic-debug-howto.rst for additional
information.
```



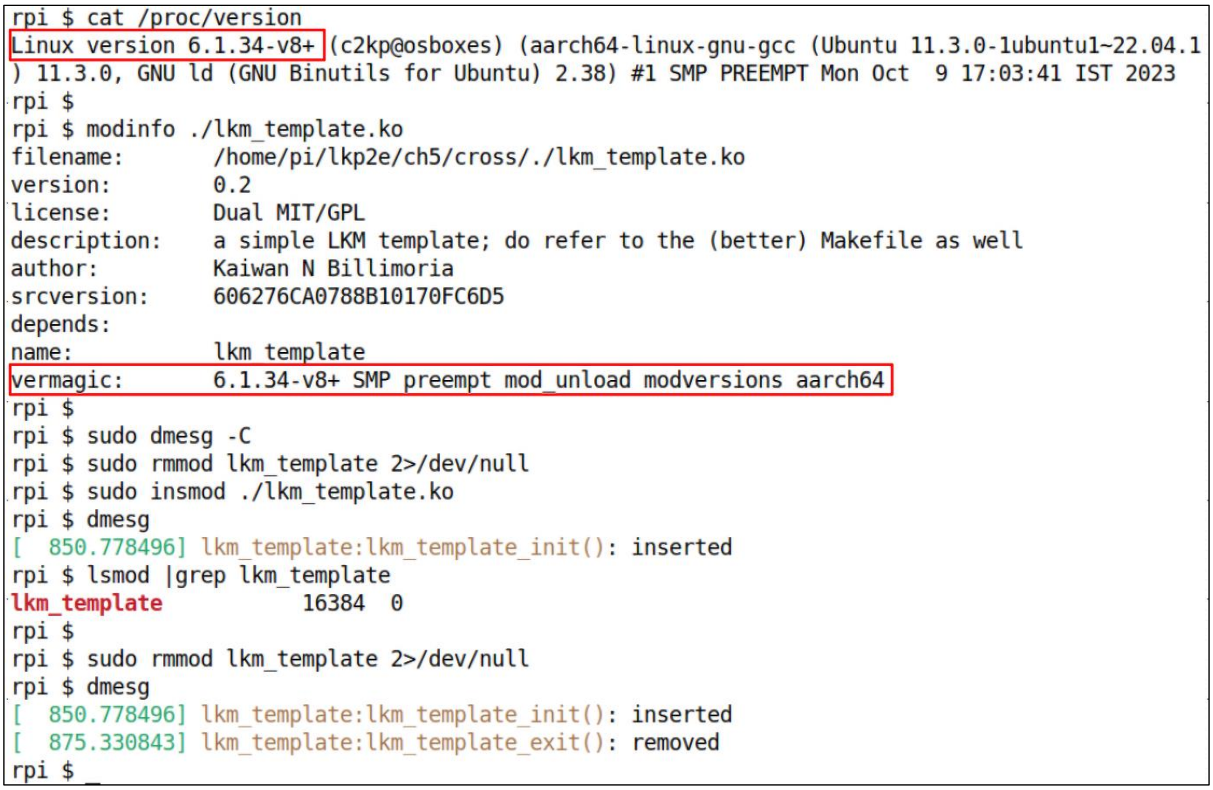
```
$ ls
Makefile  printk_loglvl.c
$ uname -r
6.1.25-lkp-kernel
$
$ ls -l /lib/modules/6.1.25-lkp-kernel/build
lrwxrwxrwx 1 root root 31 May  5 10:51 /lib/modules/6.1.25-lkp-kernel/build -> /home/c2kp/kernels
/linux-6.1.25/
$
$ make
make -C /lib/modules/6.1.25-lkp-kernel/build/ M=/home/c2kp/Linux-Kernel-Programming_2E/ch4/printk
_loglvl modules
make[1]: Entering directory '/home/c2kp/kernels/linux-6.1.25'
  CC [M]  /home/c2kp/Linux-Kernel-Programming_2E/ch4/printk_loglvl/printk_loglvl.o
  MODPOST /home/c2kp/Linux-Kernel-Programming_2E/ch4/printk_loglvl/Module.symvers
  CC [M]  /home/c2kp/Linux-Kernel-Programming_2E/ch4/printk_loglvl/printk_loglvl.mod.o
  LD [M]  /home/c2kp/Linux-Kernel-Programming_2E/ch4/printk_loglvl/printk_loglvl.ko
make[1]: Leaving directory '/home/c2kp/kernels/linux-6.1.25'
$
$ ls -a
./                  Module.symvers         printk_loglvl.mod         printk_loglvl.o
../                 .Module.symvers.cmd    printk_loglvl.mod.c       .printk_loglvl.o.cmd
Makefile            printk_loglvl.c        .printk_loglvl.mod.cmd
modules.order       printk_loglvl.ko       printk_loglvl.mod.o
.modules.order.cmd  .printk_loglvl.ko.cmd  .printk_loglvl.mod.o.cmd
$
$
$ make clean ; ls -a
make -C /lib/modules/6.1.25-lkp-kernel/build/ M=/home/c2kp/Linux-Kernel-Programming_2E/ch4/printk
_loglvl clean
make[1]: Entering directory '/home/c2kp/kernels/linux-6.1.25'
  CLEAN   /home/c2kp/Linux-Kernel-Programming_2E/ch4/printk_loglvl/Module.symvers
make[1]: Leaving directory '/home/c2kp/kernels/linux-6.1.25'
./  ../  Makefile  printk_loglvl.c
$
```



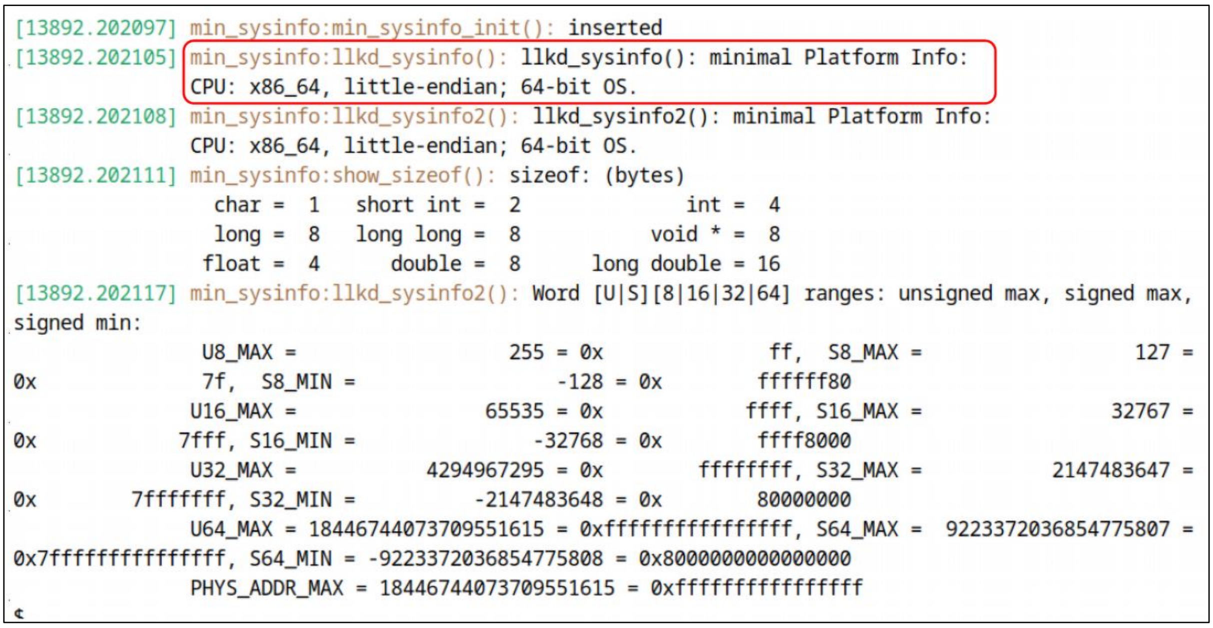
```
lkm_template $ make help
=== Makefile Help : additional targets available ===

TIP: Type make <tab><tab> to show all valid targets
FYI: KDIR=/lib/modules/6.5.6-200.fc38.x86_64/build ARCH= CROSS_COMPILE= ccflags-y="-UDEBUG -DDYNAMIC_DEBUG_MODULE" MY
DEBUG=n DBG_STRIP=n

--- usual kernel LKM targets ---
typing "make" or "all" target : builds the kernel module object (the .ko)
install     : installs the kernel module(s) to INSTALL_MOD_PATH (default here: /lib/modules/6.5.6-200.fc38.x86_64/).
            : Takes care of performing debug-only symbols stripping iff MYDEBUG=n and not using module signature
nsdeps      : namespace dependencies resolution; for possibly importing namespaces
clean       : cleanup - remove all kernel objects, temp files/dirs, etc

--- kernel code style targets ---
code-style : "wrapper" target over the following kernel code style targets
 indent     : run the indent utility on source file(s) to indent them as per the kernel code style
 checkpatch : run the kernel code style checker tool on source file(s)

--- kernel static analyzer targets ---
sa         : "wrapper" target over the following kernel static analyzer targets
 sa_sparse     : run the static analysis sparse tool on the source file(s)
 sa_gcc        : run gcc with option -W1 ("Generally useful warnings") on the source file(s)
 sa_flawfinder : run the static analysis flawfinder tool on the source file(s)
 sa_cppcheck   : run the static analysis cppcheck tool on the source file(s)
TIP: use Coccinelle as well: https://www.kernel.org/doc/html/v6.1/dev-tools/coccinelle.html

--- kernel dynamic analysis targets ---
da_kasan   : DUMMY target: this is to remind you to run your code with the dynamic analysis KASAN tool enabled; requi
res configuring the kernel with CONFIG_KASAN On, rebuild and boot it
da_lockdep : DUMMY target: this is to remind you to run your code with the dynamic analysis LOCKDEP tool (for deep lo
cking issues analysis) enabled; requires configuring the kernel with CONFIG_PROVE_LOCKING On, rebuild and boot it
TIP: Best to build a debug kernel with several kernel debug config options turned On, boot via it and run all your te
st cases

--- misc targets ---
tarxz-pkg  : tar and compress the LKM source files as a tar.xz into the dir above; allows one to transfer and build t
he module on another system
        TIP: When extracting, to extract into a directory with the same name as the tar file, do this:
              tar -xvf lkm_template.tar.xz --one-top-level
help       : this help target
```



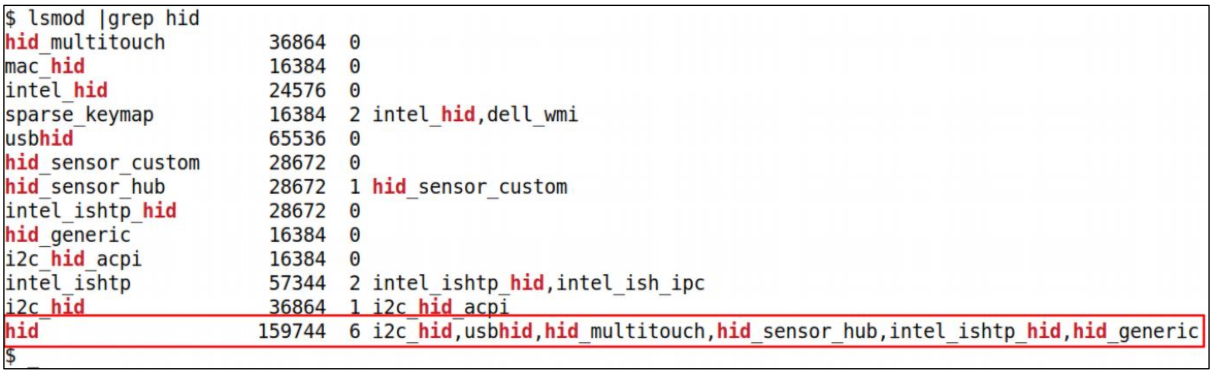
```
$ ls
lkm_template.c  Makefile  README
$ make

--- Building : KDIR=/lib/modules/6.1.25-lkp-kernel/build ARCH= CROSS_COMPILE= ccflags-y="-UDEBUG -D
DYNAMIC_DEBUG_MODULE" MYDEBUG=n DBG_STRIP=n ---
gcc (Ubuntu 11.3.0-1ubuntu1~22.04.1) 11.3.0

make -C /lib/modules/6.1.25-lkp-kernel/build M=/home/c2kp/Linux-Kernel-Programming_2E/ch5/lkm_templ
ate modules
make[1]: Entering directory '/home/c2kp/kernels/linux-6.1.25'
  CC [M]  /home/c2kp/Linux-Kernel-Programming_2E/ch5/lkm_template/lkm_template.o
  MODPOST /home/c2kp/Linux-Kernel-Programming_2E/ch5/lkm_template/Module.symvers
  CC [M]  /home/c2kp/Linux-Kernel-Programming_2E/ch5/lkm_template/lkm_template.mod.o
  LD [M]  /home/c2kp/Linux-Kernel-Programming_2E/ch5/lkm_template/lkm_template.ko
make[1]: Leaving directory '/home/c2kp/kernels/linux-6.1.25'
if [ "n" = "y" ]; then \
   strip --strip-debug lkm_template.ko ; \
fi
$ ls -lh ./lkm_template.ko
-rw-rw-r-- 1 c2kp c2kp 108K Oct 14 10:36 ./lkm_template.ko
$
$ sudo dmesg -C
$ sudo insmod ./lkm_template.ko
$ lsmod |head -n2
Module                  Size  Used by
lkm_template           16384  0
$
$ sudo rmmod lkm_template
$ sudo dmesg
[ 2012.653246] lkm_template:lkm_template_init(): inserted
[ 2029.253820] lkm_template:lkm_template_exit(): removed
$
```



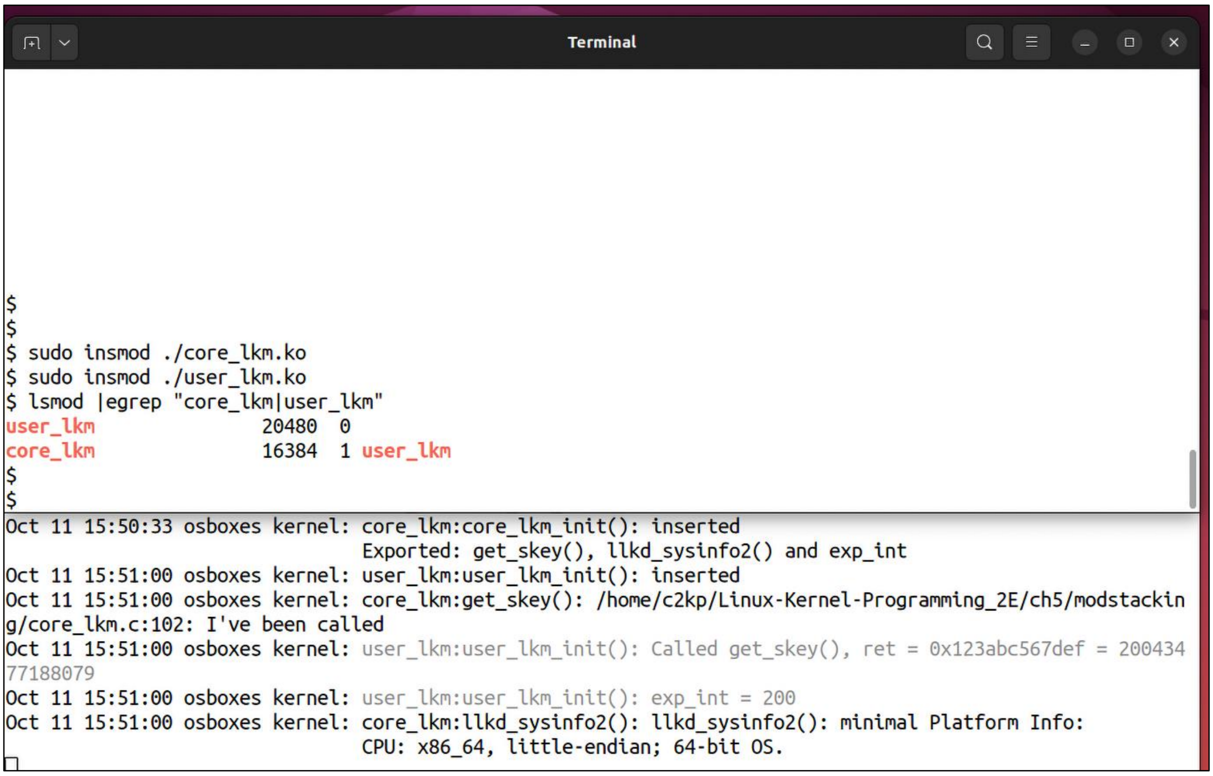
```
rpi $ cat /proc/version
Linux version 6.1.34-v8+ (c2kp@osboxes) (aarch64-linux-gnu-gcc (Ubuntu 11.3.0-1ubuntu1~22.04.1
) 11.3.0, GNU ld (GNU Binutils for Ubuntu) 2.38) #1 SMP PREEMPT Mon Oct  9 17:03:41 IST 2023
rpi $
rpi $ modinfo ./lkm_template.ko
filename:       /home/pi/lkp2e/ch5/cross/./lkm_template.ko
version:        0.2
license:        Dual MIT/GPL
description:    a simple LKM template; do refer to the (better) Makefile as well
author:         Kaiwan N Billimoria
srcversion:     606276CA0788B10170FC6D5
depends:
name:           lkm_template
vermagic:       6.1.34-v8+ SMP preempt mod_unload modversions aarch64
rpi $
rpi $ sudo dmesg -C
rpi $ sudo rmmod lkm_template 2>/dev/null
rpi $ sudo insmod ./lkm_template.ko
rpi $ dmesg
[  850.778496] lkm_template:lkm_template_init(): inserted
rpi $ lsmod |grep lkm_template
lkm_template           16384  0
rpi $
rpi $ sudo rmmod lkm_template 2>/dev/null
rpi $ dmesg
[  850.778496] lkm_template:lkm_template_init(): inserted
[  875.330843] lkm_template:lkm_template_exit(): removed
rpi $ _
```



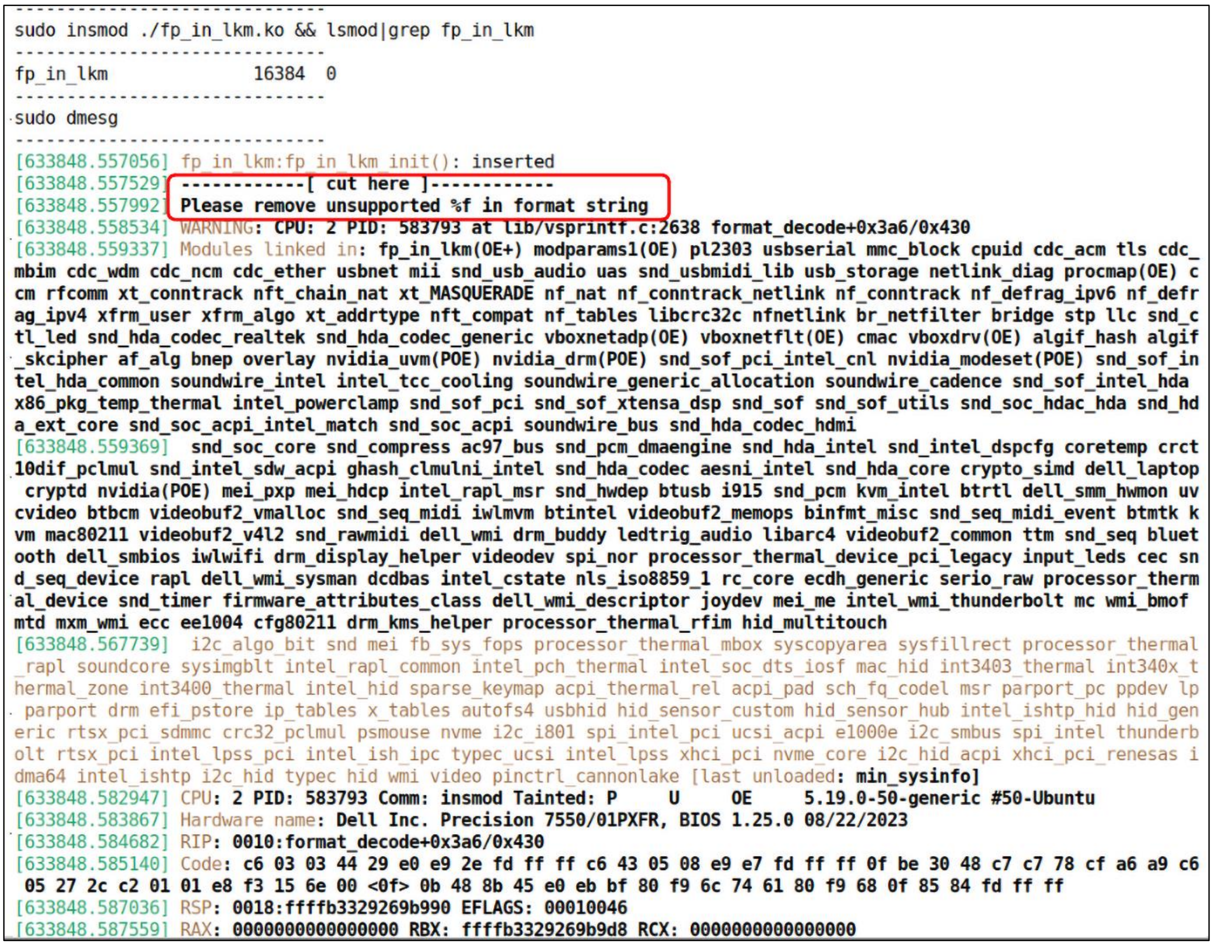
```
[13892.202097] min_sysinfo:min_sysinfo_init(): inserted
[13892.202105] min_sysinfo:llkd_sysinfo(): llkd_sysinfo(): minimal Platform Info:
               CPU: x86_64, little-endian; 64-bit OS.
[13892.202108] min_sysinfo:llkd_sysinfo2(): llkd_sysinfo2(): minimal Platform Info:
               CPU: x86_64, little-endian; 64-bit OS.
[13892.202111] min_sysinfo:show_sizeof(): sizeof: (bytes)
                 char =  1   short int =  2              int =  4
                 long =  8   long long =  8           void * =  8
                float =  4      double =  8      long double = 16
[13892.202117] min_sysinfo:llkd_sysinfo2(): Word [U|S][8|16|32|64] ranges: unsigned max, signed max,
signed min:
                U8_MAX =                  255 = 0x              ff,  S8_MAX =                   127 =
0x              7f,  S8_MIN =                  -128 = 0x        ffffff80
               U16_MAX =                65535 = 0x            ffff, S16_MAX =                 32767 =
0x            7fff, S16_MIN =                -32768 = 0x        ffff8000
               U32_MAX =           4294967295 = 0x        ffffffff, S32_MAX =            2147483647 =
0x        7fffffff, S32_MIN =           -2147483648 = 0x        80000000
               U64_MAX = 18446744073709551615 = 0xffffffffffffffff, S64_MAX =  9223372036854775807 =
0x7fffffffffffffff, S64_MIN = -9223372036854775808 = 0x8000000000000000
               PHYS_ADDR_MAX = 18446744073709551615 = 0xffffffffffffffff
¢
```



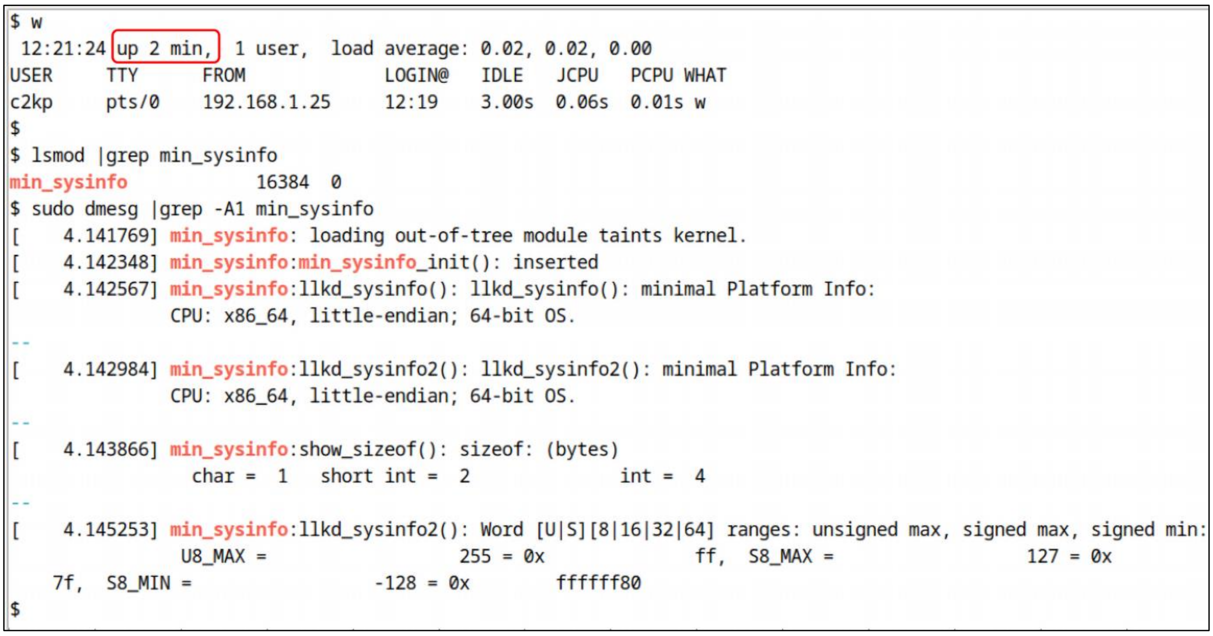
```
$ lsmod |grep hid
hid_multitouch         36864  0
mac_hid                16384  0
intel_hid              24576  0
sparse_keymap          16384  2 intel_hid,dell_wmi
usbhid                 65536  0
hid_sensor_custom      28672  0
hid_sensor_hub         28672  1 hid_sensor_custom
intel_ishtp_hid        28672  0
hid_generic            16384  0
i2c_hid_acpi           16384  0
intel_ishtp            57344  2 intel_ishtp_hid,intel_ish_ipc
i2c_hid                36864  1 i2c_hid_acpi
hid                   159744  6 i2c_hid,usbhid,hid_multitouch,hid_sensor_hub,intel_ishtp_hid,hid_generic
$ _
```



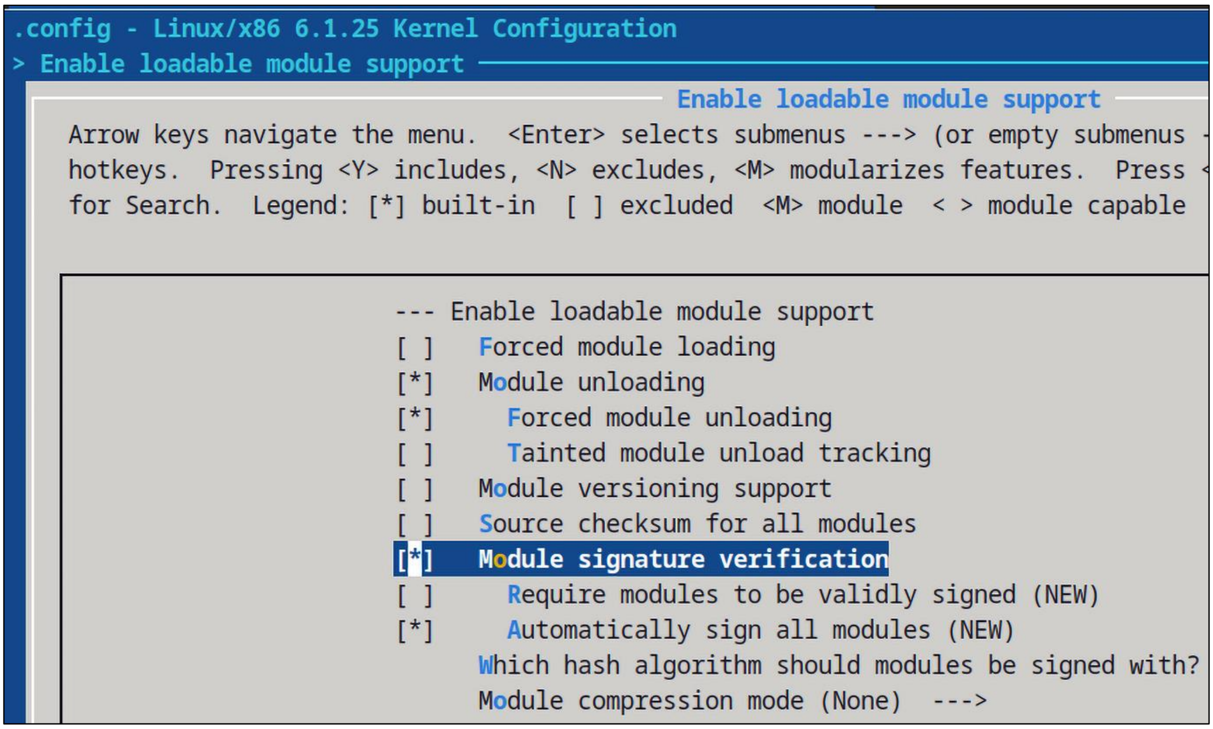
```
$
$
$ sudo insmod ./core_lkm.ko
$ sudo insmod ./user_lkm.ko
$ lsmod |egrep "core_lkm|user_lkm"
user_lkm               20480  0
core_lkm               16384  1 user_lkm
$
$
Oct 11 15:50:33 osboxes kernel: core_lkm:core_lkm_init(): inserted
                                Exported: get_skey(), llkd_sysinfo2() and exp_int
Oct 11 15:51:00 osboxes kernel: user_lkm:user_lkm_init(): inserted
Oct 11 15:51:00 osboxes kernel: core_lkm:get_skey(): /home/c2kp/Linux-Kernel-Programming_2E/ch5/modstackin
g/core_lkm.c:102: I've been called
Oct 11 15:51:00 osboxes kernel: user_lkm:user_lkm_init(): Called get_skey(), ret = 0x123abc567def = 200434
77188079
Oct 11 15:51:00 osboxes kernel: user_lkm:user_lkm_init(): exp_int = 200
Oct 11 15:51:00 osboxes kernel: core_lkm:llkd_sysinfo2(): llkd_sysinfo2(): minimal Platform Info:
                                CPU: x86_64, little-endian; 64-bit OS.
```



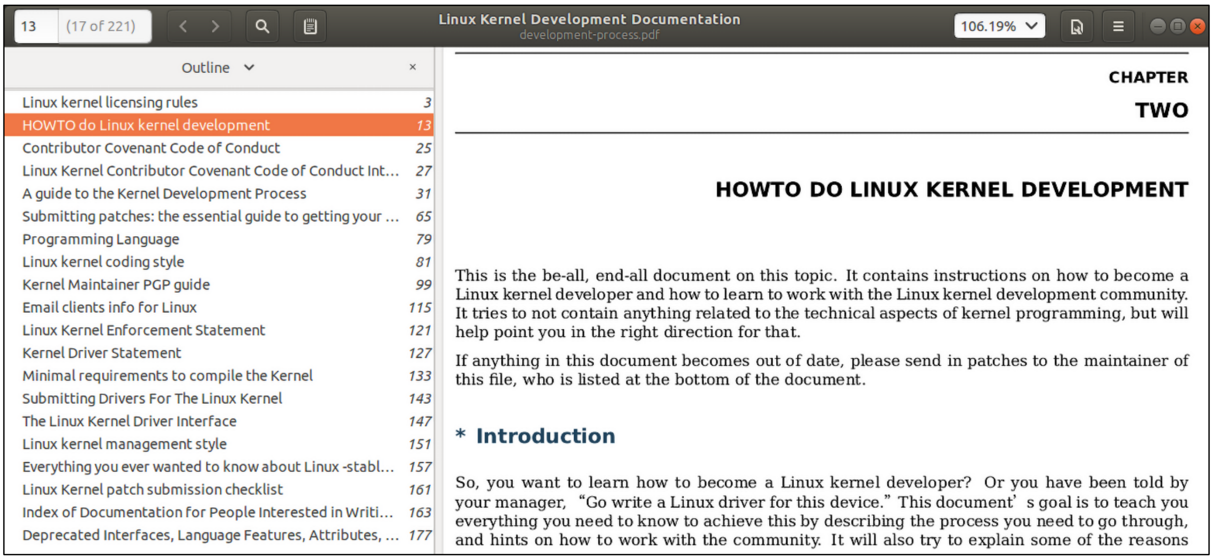
```
-----------------------------
sudo insmod ./fp_in_lkm.ko && lsmod|grep fp_in_lkm
-----------------------------
fp_in_lkm              16384  0
-----------------------------
sudo dmesg
-----------------------------
[633848.557056] fp_in_lkm:fp_in_lkm_init(): inserted
[633848.557529] ------------[ cut here ]------------
[633848.557992] Please remove unsupported %f in format string
[633848.558534] WARNING: CPU: 2 PID: 583793 at lib/vsprintf.c:2638 format_decode+0x3a6/0x430
[633848.559337] Modules linked in: fp_in_lkm(OE+) modparams1(OE) pl2303 usbserial mmc_block cpuid cdc_acm tls cdc_
mbim cdc_wdm cdc_ncm cdc_ether usbnet mii snd_usb_audio uas snd_usbmidi_lib usb_storage netlink_diag procmap(OE) c
cm rfcomm xt_conntrack nft_chain_nat xt_MASQUERADE nf_nat nf_conntrack_netlink nf_conntrack nf_defrag_ipv6 nf_defr
ag_ipv4 xfrm_user xfrm_algo xt_addrtype nft_compat nf_tables libcrc32c nfnetlink br_netfilter bridge stp llc snd_c
tl_led snd_hda_codec_realtek snd_hda_codec_generic vboxnetadp(OE) vboxnetflt(OE) cmac vboxdrv(OE) algif_hash algif
_skcipher af_alg bnep overlay nvidia_uvm(POE) nvidia_drm(POE) snd_sof_pci_intel_cnl nvidia_modeset(POE) snd_sof_in
tel_hda_common soundwire_intel intel_tcc_cooling soundwire_generic_allocation soundwire_cadence snd_sof_intel_hda
x86_pkg_temp_thermal intel_powerclamp snd_sof_pci snd_sof_xtensa_dsp snd_sof snd_sof_utils snd_soc_hdac_hda snd_hd
a_ext_core snd_soc_acpi_intel_match snd_soc_acpi soundwire_bus snd_hda_codec_hdmi
[633848.559369]  snd_soc_core snd_compress ac97_bus snd_pcm_dmaengine snd_hda_intel snd_intel_dspcfg coretemp crct
10dif_pclmul snd_intel_sdw_acpi ghash_clmulni_intel snd_hda_codec aesni_intel snd_hda_core crypto_simd dell_laptop
 cryptd nvidia(POE) mei_pxp mei_hdcp intel_rapl_msr snd_hwdep btusb i915 snd_pcm kvm_intel btrtl dell_smm_hwmon uv
cvideo btbcm videobuf2_vmalloc snd_seq_midi iwlmvm btintel videobuf2_memops binfmt_misc snd_seq_midi_event btmtk k
vm mac80211 videobuf2_v4l2 snd_rawmidi dell_wmi drm_buddy ledtrig_audio libarc4 videobuf2_common ttm snd_seq bluet
ooth dell_smbios iwlwifi drm_display_helper videodev spi_nor processor_thermal_device_pci_legacy input_leds cec sn
d_seq_device rapl dell_wmi_sysman dcdbas intel_cstate nls_iso8859_1 rc_core ecdh_generic serio_raw processor_therm
al_device snd_timer firmware_attributes_class dell_wmi_descriptor joydev mei_me intel_wmi_thunderbolt mc wmi_bmof
mtd mxm_wmi ecc ee1004 cfg80211 drm_kms_helper processor_thermal_rfim hid_multitouch
[633848.567739]  i2c_algo_bit snd mei fb_sys_fops processor_thermal_mbox syscopyarea sysfillrect processor_thermal
_rapl soundcore sysimgblt intel_rapl_common intel_pch_thermal intel_soc_dts_iosf mac_hid int3403_thermal int340x_t
hermal_zone int3400_thermal intel_hid sparse_keymap acpi_thermal_rel acpi_pad sch_fq_codel msr parport_pc ppdev lp
 parport drm efi_pstore ip_tables x_tables autofs4 usbhid hid_sensor_custom hid_sensor_hub intel_ishtp_hid hid_gen
eric rtsx_pci_sdmmc crc32_pclmul psmouse nvme i2c_i801 spi_intel_pci ucsi_acpi e1000e i2c_smbus spi_intel thunderb
olt rtsx_pci intel_lpss_pci intel_ish_ipc typec_ucsi intel_lpss xhci_pci nvme_core i2c_hid_acpi xhci_pci_renesas i
dma64 intel_ishtp i2c_hid typec hid wmi video pinctrl_cannonlake [last unloaded: min_sysinfo]
[633848.582947] CPU: 2 PID: 583793 Comm: insmod Tainted: P     U     OE     5.19.0-50-generic #50-Ubuntu
[633848.583867] Hardware name: Dell Inc. Precision 7550/01PXFR, BIOS 1.25.0 08/22/2023
[633848.584682] RIP: 0010:format_decode+0x3a6/0x430
[633848.585140] Code: c6 03 03 44 29 e0 e9 2e fd ff ff c6 43 05 08 e9 e7 fd ff ff 0f be 30 48 c7 c7 78 cf a6 a9 c6
 05 27 2c c2 01 01 e8 f3 15 6e 00 <0f> 0b 48 8b 45 e0 eb bf 80 f9 6c 74 61 80 f9 68 0f 85 84 fd ff ff
[633848.587036] RSP: 0018:ffffb3329269b990 EFLAGS: 00010046
[633848.587559] RAX: 0000000000000000 RBX: ffffb3329269b9d8 RCX: 0000000000000000
```



```
$ w
 12:21:24 up 2 min,  1 user,  load average: 0.02, 0.02, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
c2kp     pts/0    192.168.1.25     12:19    3.00s  0.06s  0.01s w
$
$ lsmod |grep min_sysinfo
min_sysinfo            16384  0
$ sudo dmesg |grep -A1 min_sysinfo
[    4.141769] min_sysinfo: loading out-of-tree module taints kernel.
[    4.142348] min_sysinfo:min_sysinfo_init(): inserted
[    4.142567] min_sysinfo:llkd_sysinfo(): llkd_sysinfo(): minimal Platform Info:
               CPU: x86_64, little-endian; 64-bit OS.
--
[    4.142984] min_sysinfo:llkd_sysinfo2(): llkd_sysinfo2(): minimal Platform Info:
               CPU: x86_64, little-endian; 64-bit OS.
--
[    4.143866] min_sysinfo:show_sizeof(): sizeof: (bytes)
                 char =  1   short int =  2           int =  4
--
[    4.145253] min_sysinfo:llkd_sysinfo2(): Word [U|S][8|16|32|64] ranges: unsigned max, signed max, signed min:
                U8_MAX =                 255 = 0x          ff,  S8_MAX =                 127 = 0x
    7f,  S8_MIN =                -128 = 0x       ffffff80
$
```



```
.config - Linux/x86 6.1.25 Kernel Configuration
> Enable loadable module support ─────────────────────────────────────────────
  ┌──────────────────────── Enable loadable module support ────────
  │  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus -
  │  hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <
  │  for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable
  │
  │ ┌──────────────────────────────────────────────────────────────
  │ │                   --- Enable loadable module support
  │ │                   [ ]   Forced module loading
  │ │                   [*]   Module unloading
  │ │                   [*]     Forced module unloading
  │ │                   [ ]     Tainted module unload tracking
  │ │                   [ ]   Module versioning support
  │ │                   [ ]   Source checksum for all modules
  │ │                   [*]   Module signature verification
  │ │                   [ ]     Require modules to be validly signed (NEW)
  │ │                   [*]     Automatically sign all modules (NEW)
  │ │                         Which hash algorithm should modules be signed with?
  │ │                         Module compression mode (None)  --->
```



CHAPTER

TWO

HOWTO DO LINUX KERNEL DEVELOPMENT

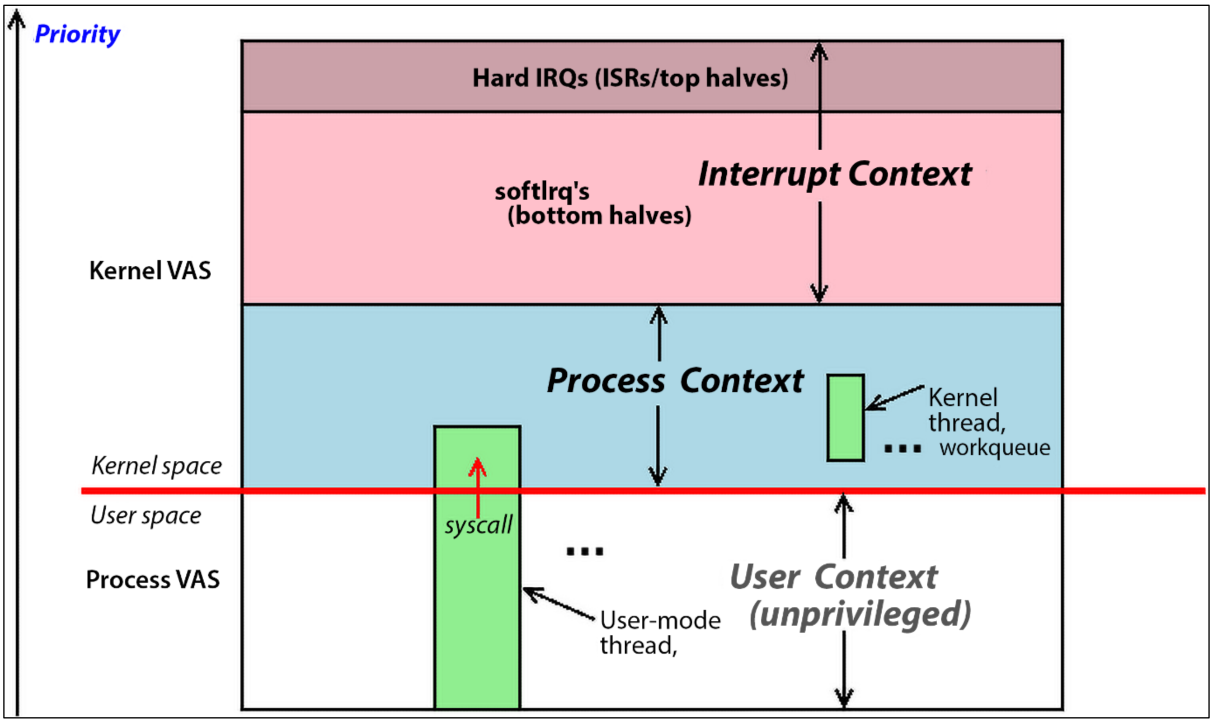
This is the be-all, end-all document on this topic. It contains instructions on how to become a Linux kernel developer and how to learn to work with the Linux kernel development community. It tries to not contain anything related to the technical aspects of kernel programming, but will help point you in the right direction for that.

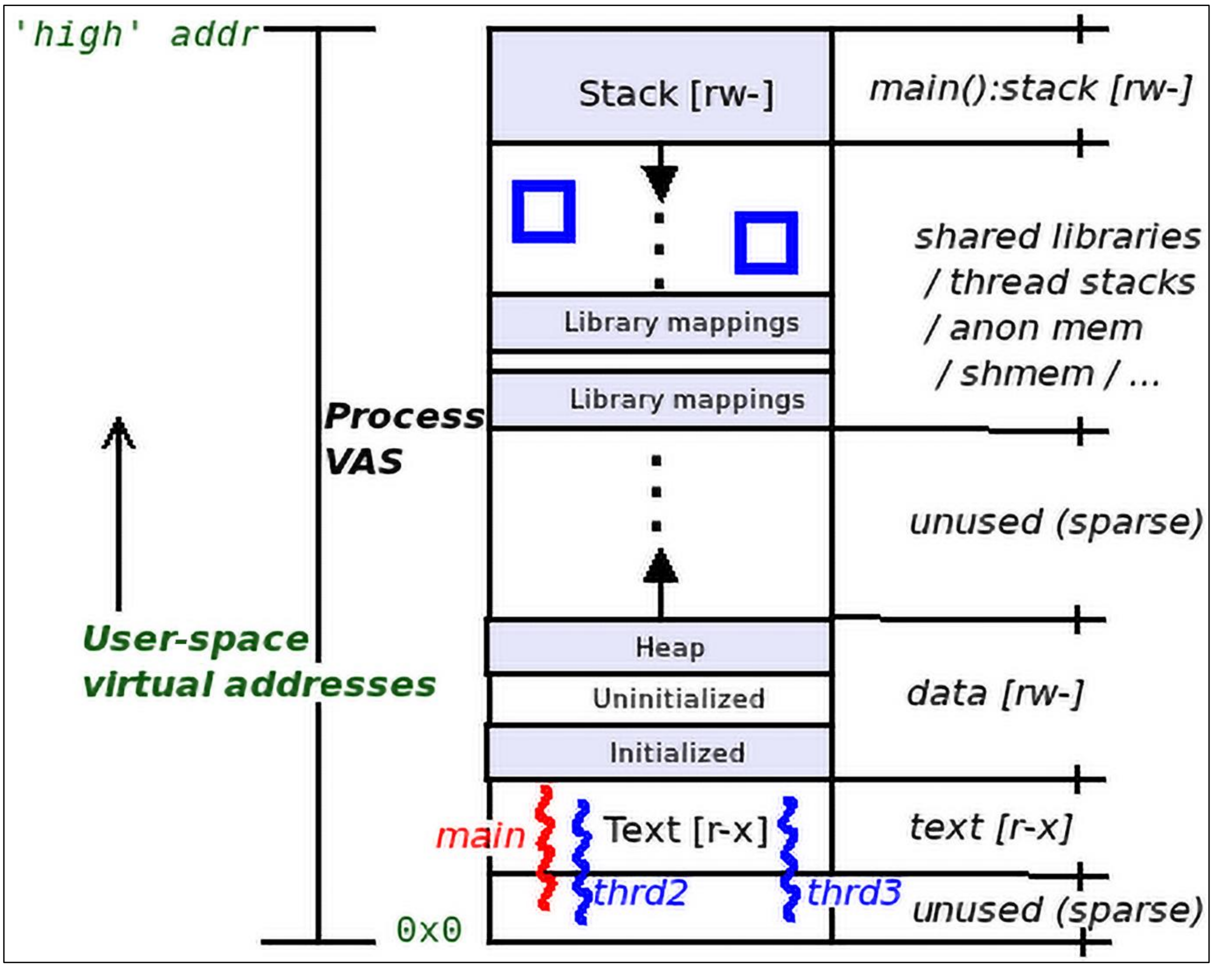
If anything in this document becomes out of date, please send in patches to the maintainer of this file, who is listed at the bottom of the document.

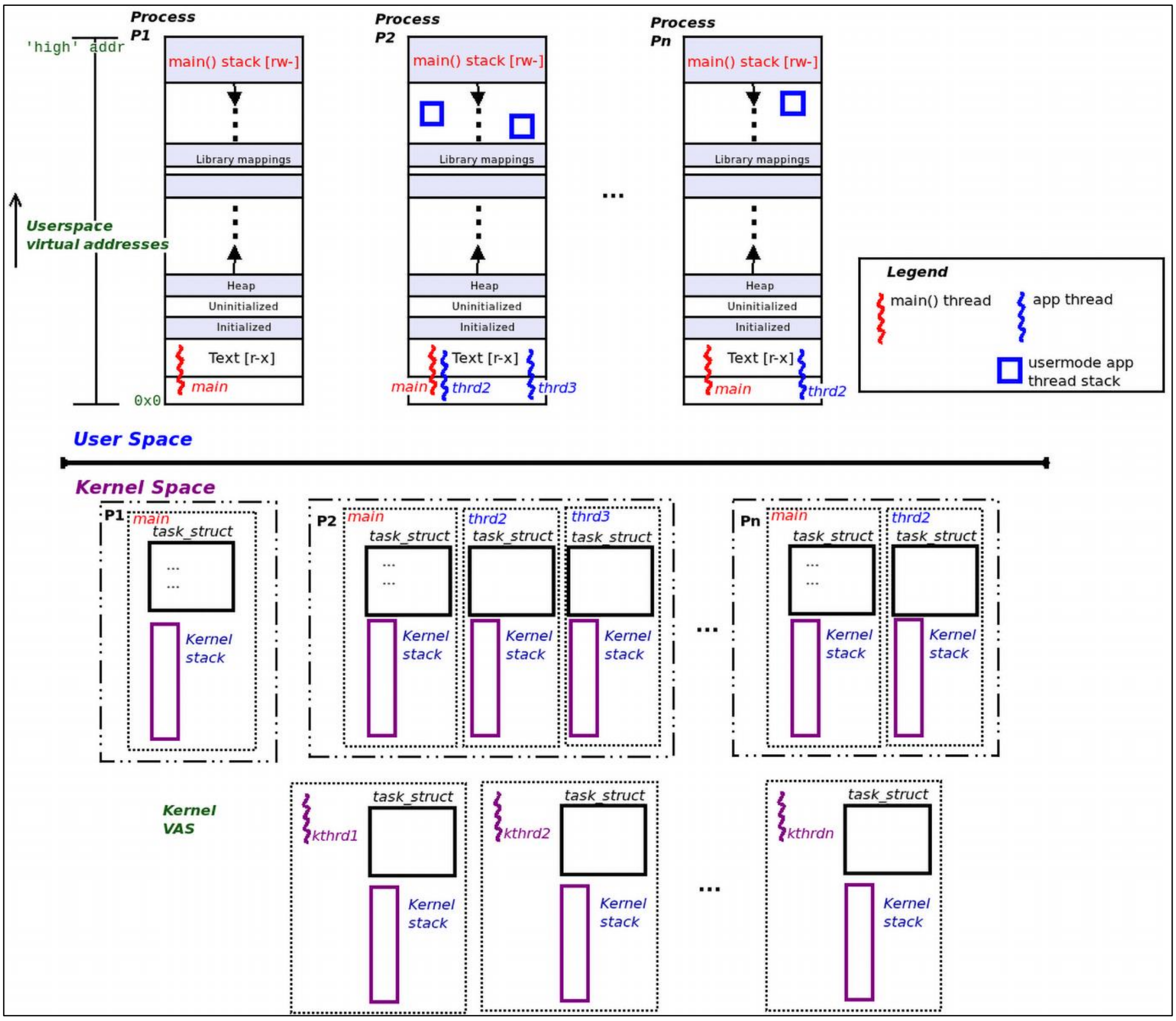
* Introduction

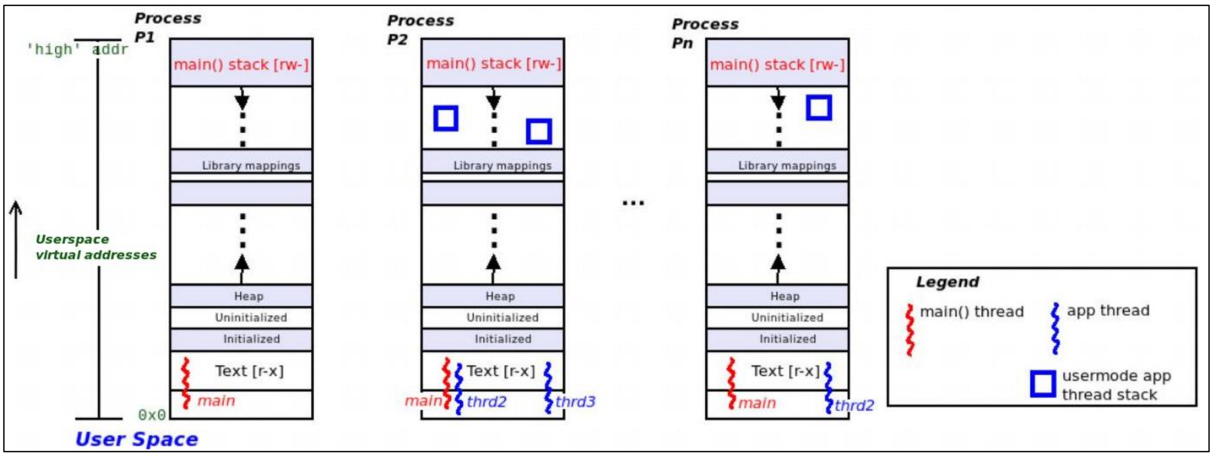
So, you want to learn how to become a Linux kernel developer? Or you have been told by your manager, "Go write a Linux driver for this device." This document' s goal is to teach you everything you need to know to achieve this by describing the process you need to go through, and hints on how to work with the community. It will also try to explain some of the reasons

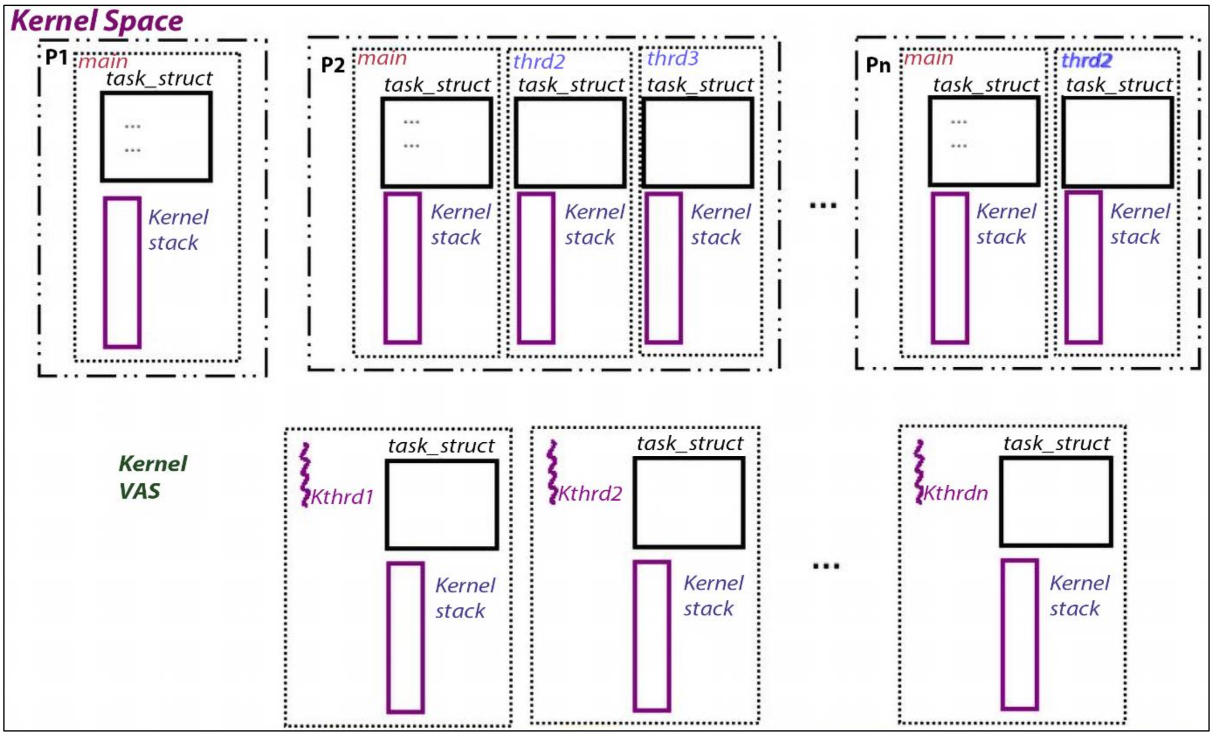
Chapter 6: Kernel Internals Essentials – Processes and Threads

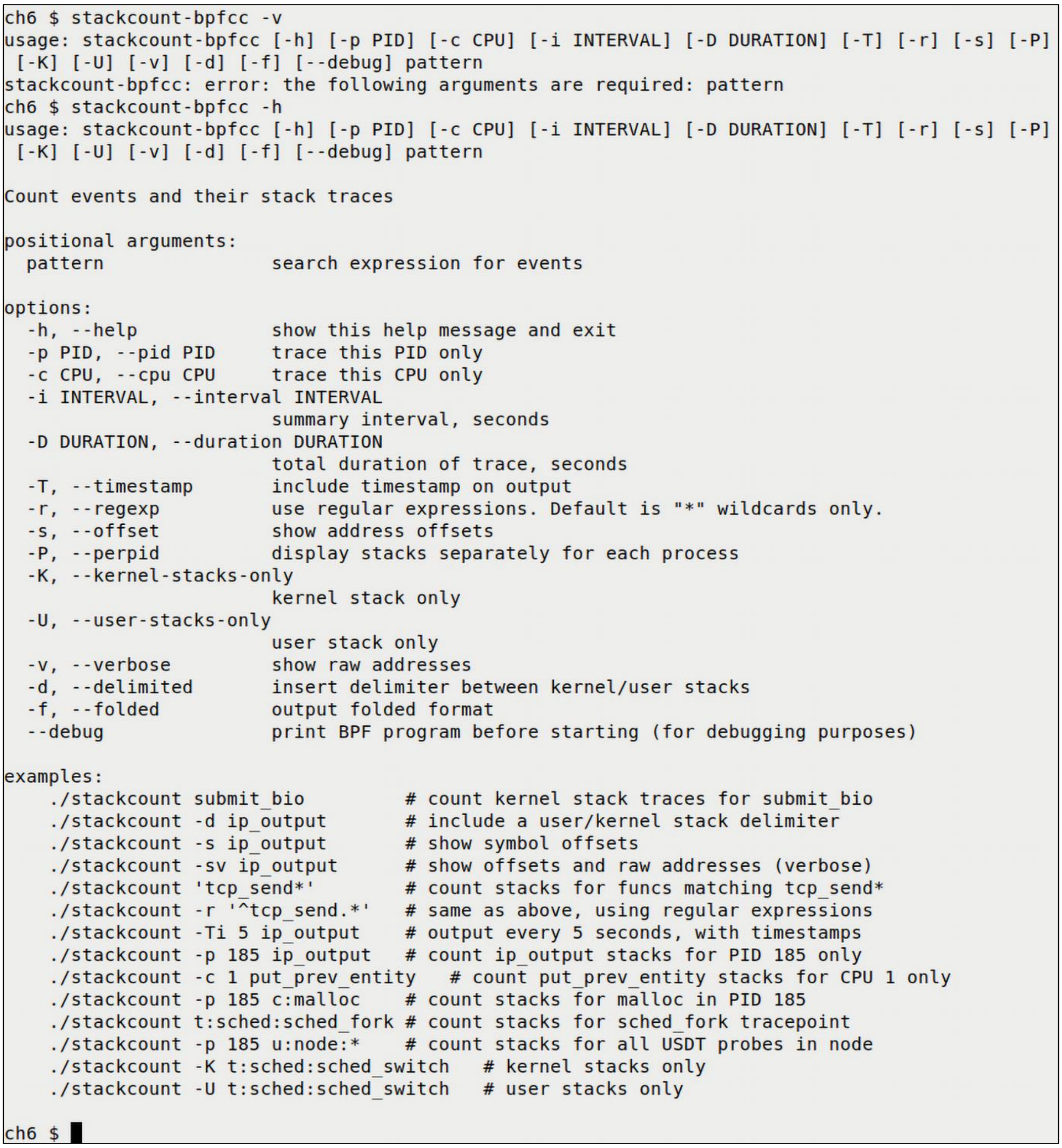



'high' addr
Stack [rw-]
main():stack [rw-]
Library mappings
Library mappings
shared libraries / thread stacks / anon mem / shmem / ...
Process VAS
unused (sparse)
User-space virtual addresses
Heap
Uninitialized
Initialized
data [rw-]
main
Text [r-x]
text [r-x]
thrd2
thrd3
unused (sparse)
0x0

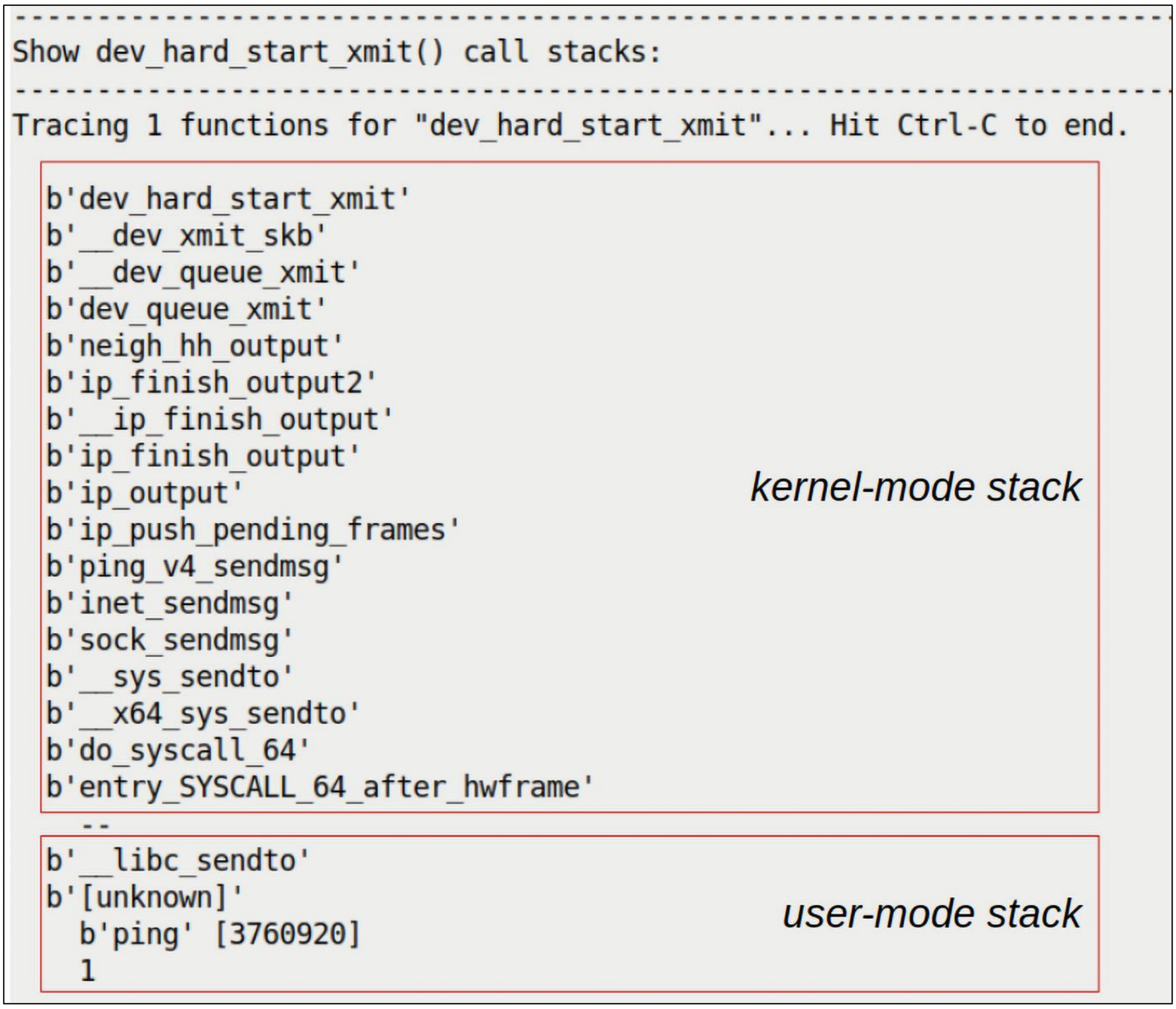

'high' addr
Userspace virtual addresses
0x0
Process P1
Process P2
Process Pn
main() stack [rw-]
Library mappings
Heap
Uninitialized
Initialized
Text [r-x]
main
thrd2
thrd3
...
Legend
main() thread
app thread
usermode app thread stack
User Space
Kernel Space
P1
P2
Pn
task_struct
Kernel stack
Kernel VAS
kthrd1
kthrd2
kthrdn


'high' addr
Userspace virtual addresses
0x0
Process P1
Process P2
Process Pn
main() stack [rw-]
Library mappings
Heap
Uninitialized
Initialized
Text [r-x]
main
thrd2
thrd3
...
Legend
main() thread
app thread
usermode app thread stack
User Space

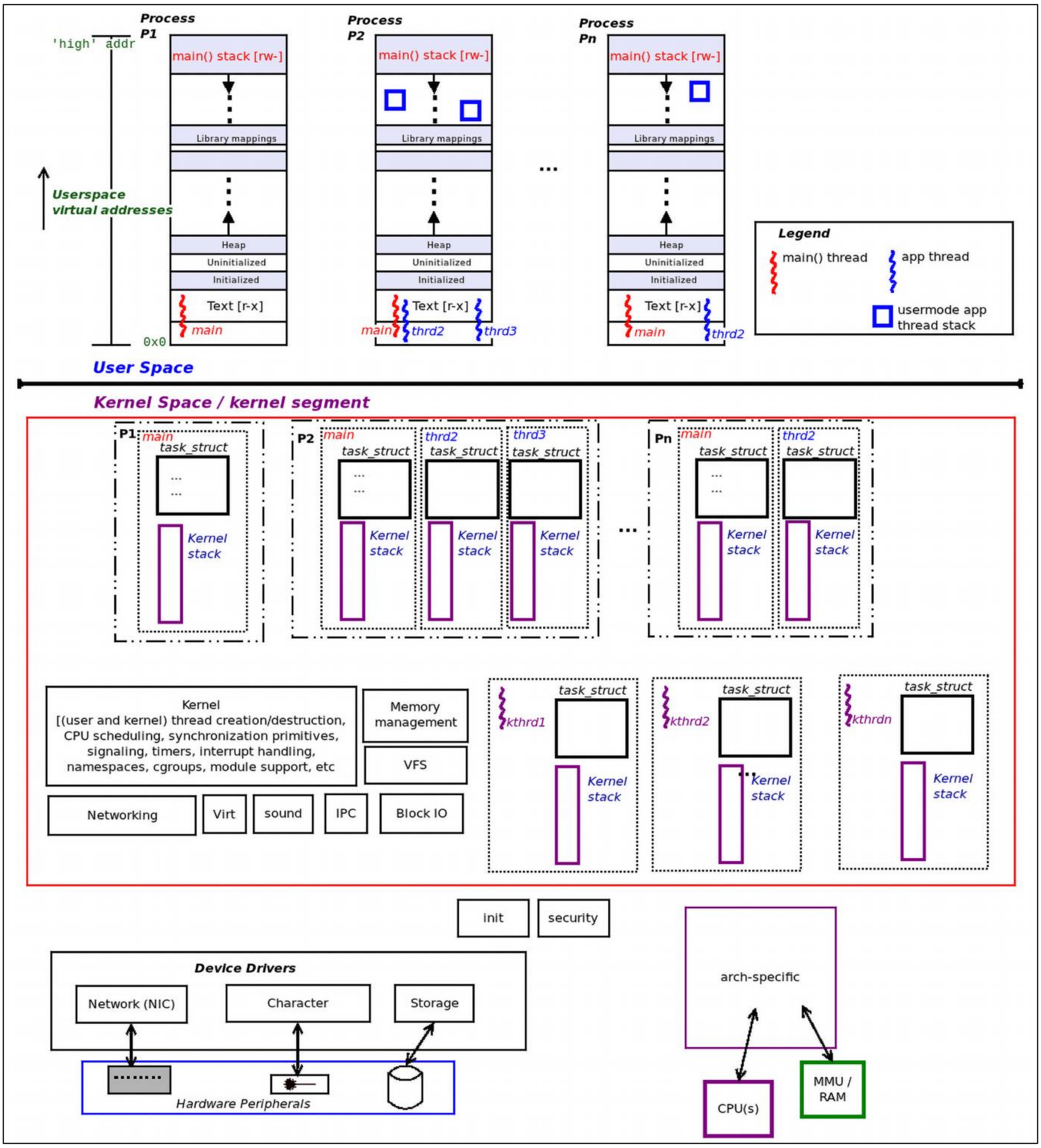

Kernel Space
P1
main
task_struct
...
...
Kernel stack
P2
main
task_struct
Kernel stack
thrd2
task_struct
Kernel stack
thrd3
task_struct
Kernel stack
...
Pn
main
task_struct
Kernel stack
thrd2
task_struct
Kernel stack
Kernel VAS
Kthrd1
task_struct
Kernel stack
Kthrd2
task_struct
Kernel stack
...
Kthrdn
task_struct
Kernel stack

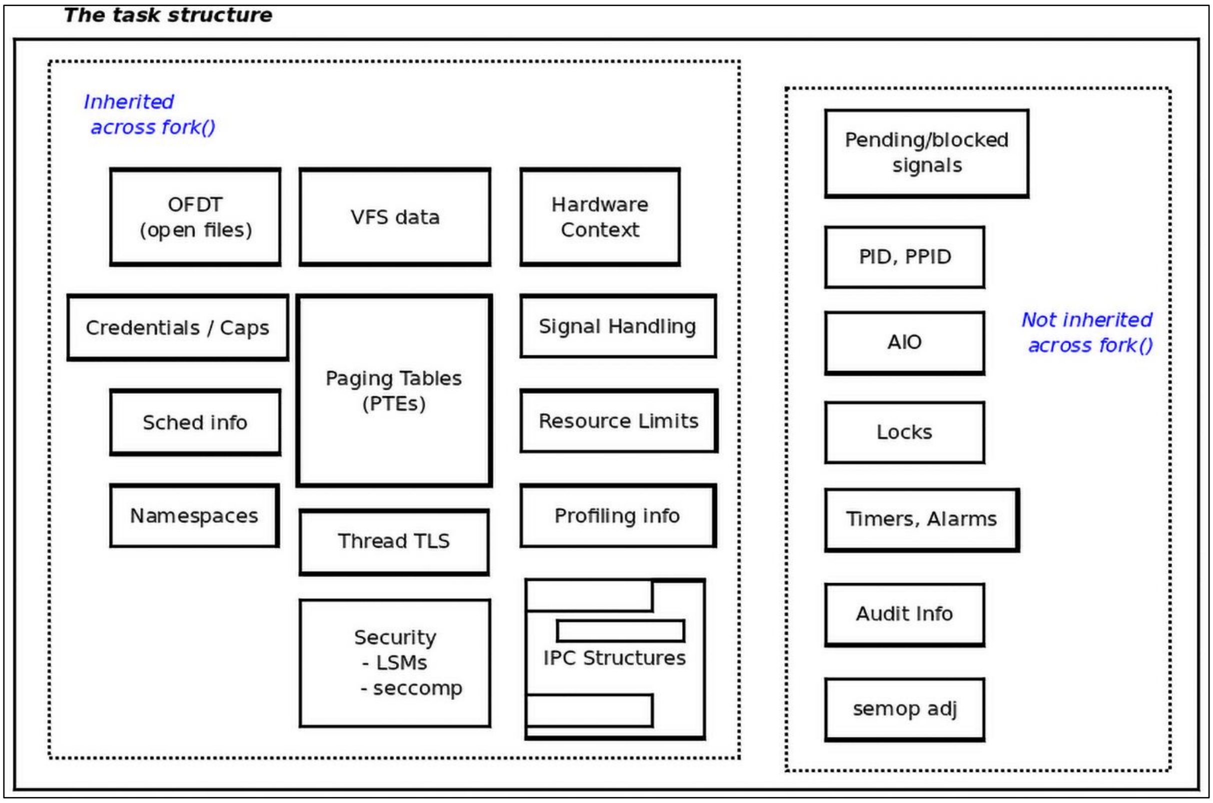


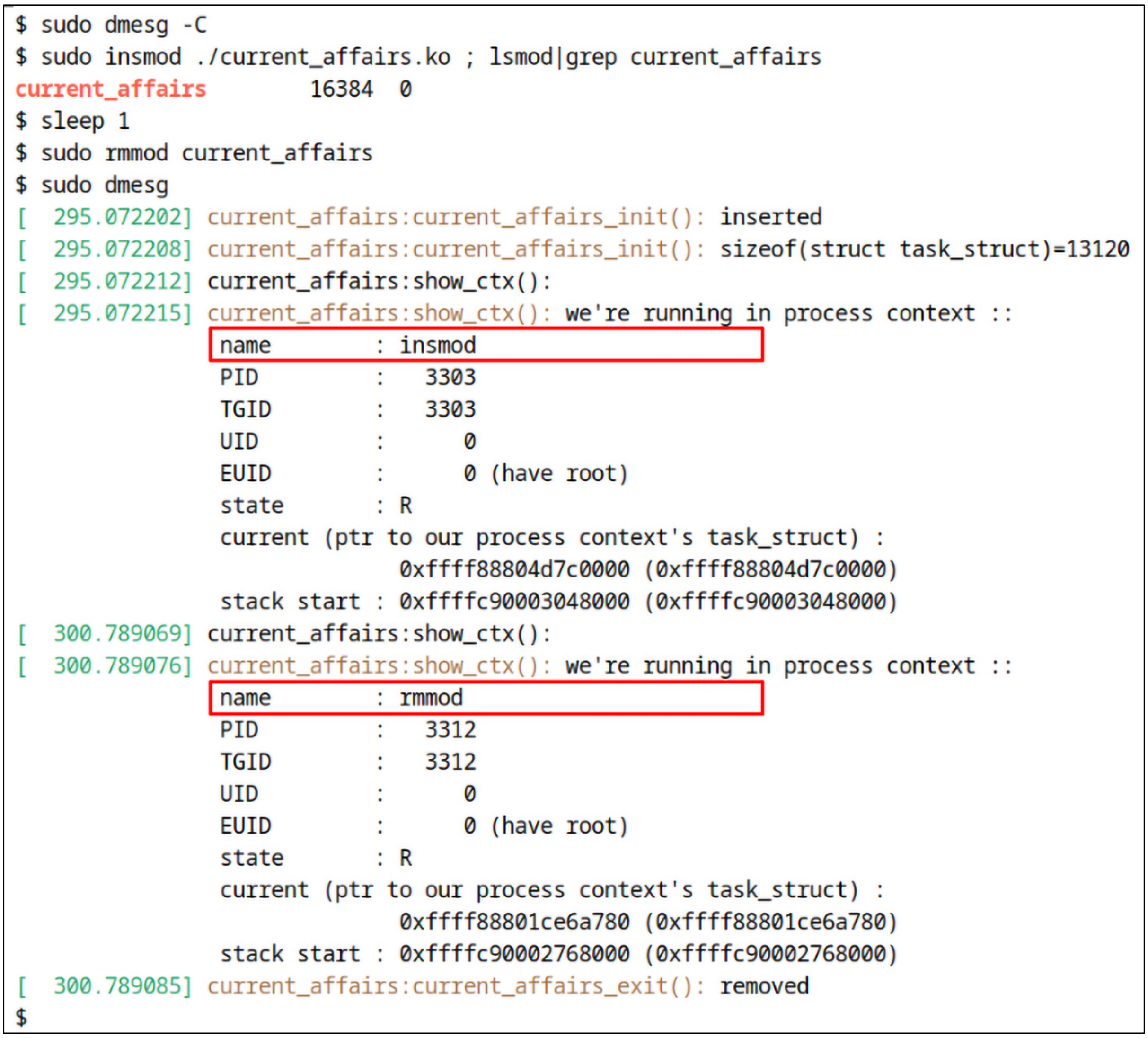
```
ch6 $ stackcount-bpfcc -v
usage: stackcount-bpfcc [-h] [-p PID] [-c CPU] [-i INTERVAL] [-D DURATION] [-T] [-r] [-s] [-P]
 [-K] [-U] [-v] [-d] [-f] [--debug] pattern
stackcount-bpfcc: error: the following arguments are required: pattern
ch6 $ stackcount-bpfcc -h
usage: stackcount-bpfcc [-h] [-p PID] [-c CPU] [-i INTERVAL] [-D DURATION] [-T] [-r] [-s] [-P]
 [-K] [-U] [-v] [-d] [-f] [--debug] pattern

Count events and their stack traces

positional arguments:
  pattern               search expression for events

options:
  -h, --help            show this help message and exit
  -p PID, --pid PID     trace this PID only
  -c CPU, --cpu CPU     trace this CPU only
  -i INTERVAL, --interval INTERVAL
                        summary interval, seconds
  -D DURATION, --duration DURATION
                        total duration of trace, seconds
  -T, --timestamp       include timestamp on output
  -r, --regexp          use regular expressions. Default is "*" wildcards only.
  -s, --offset          show address offsets
  -P, --perpid          display stacks separately for each process
  -K, --kernel-stacks-only
                        kernel stack only
  -U, --user-stacks-only
                        user stack only
  -v, --verbose         show raw addresses
  -d, --delimited       insert delimiter between kernel/user stacks
  -f, --folded          output folded format
  --debug               print BPF program before starting (for debugging purposes)

examples:
    ./stackcount submit_bio         # count kernel stack traces for submit_bio
    ./stackcount -d ip_output       # include a user/kernel stack delimiter
    ./stackcount -s ip_output       # show symbol offsets
    ./stackcount -sv ip_output      # show offsets and raw addresses (verbose)
    ./stackcount 'tcp_send*'        # count stacks for funcs matching tcp_send*
    ./stackcount -r '^tcp_send.*'   # same as above, using regular expressions
    ./stackcount -Ti 5 ip_output    # output every 5 seconds, with timestamps
    ./stackcount -p 185 ip_output   # count ip_output stacks for PID 185 only
    ./stackcount -c 1 put_prev_entity   # count put_prev_entity stacks for CPU 1 only
    ./stackcount -p 185 c:malloc    # count stacks for malloc in PID 185
    ./stackcount t:sched:sched_fork # count stacks for sched_fork tracepoint
    ./stackcount -p 185 u:node:*    # count stacks for all USDT probes in node
    ./stackcount -K t:sched:sched_switch   # kernel stacks only
    ./stackcount -U t:sched:sched_switch   # user stacks only

ch6 $ █
```



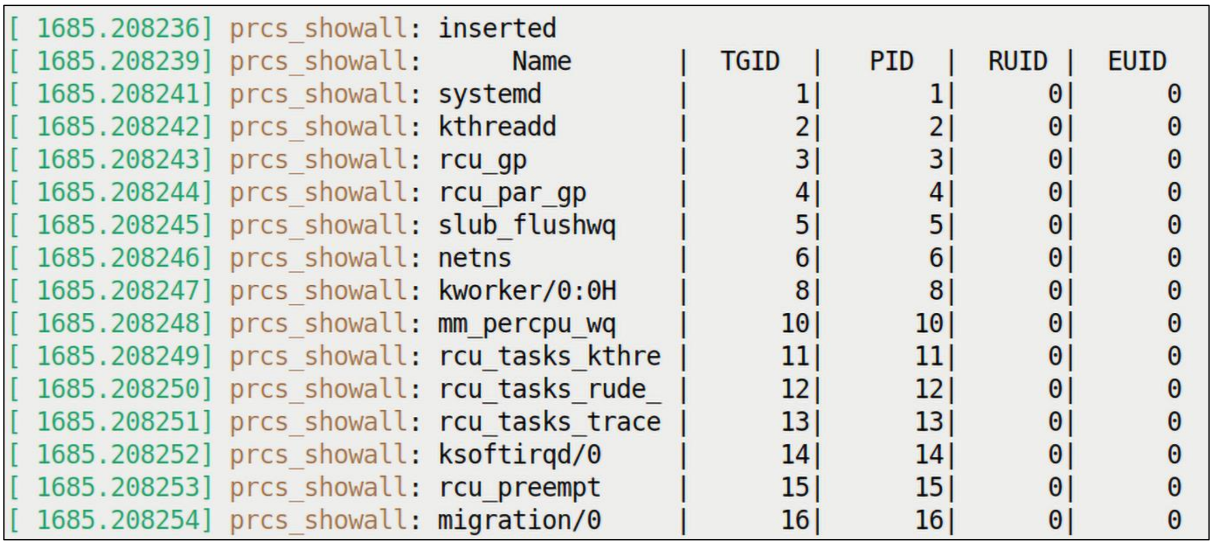
```
--------------------------------------------------------------------
Show dev_hard_start_xmit() call stacks:
--------------------------------------------------------------------
Tracing 1 functions for "dev_hard_start_xmit"... Hit Ctrl-C to end.
  b'dev_hard_start_xmit'
  b'__dev_xmit_skb'
  b'__dev_queue_xmit'
  b'dev_queue_xmit'
  b'neigh_hh_output'
  b'ip_finish_output2'
  b'__ip_finish_output'
  b'ip_finish_output'
  b'ip_output'
  b'ip_push_pending_frames'
  b'ping_v4_sendmsg'
  b'inet_sendmsg'
  b'sock_sendmsg'
  b'__sys_sendto'
  b'__x64_sys_sendto'
  b'do_syscall_64'
  b'entry_SYSCALL_64_after_hwframe'
    --
  b'__libc_sendto'
  b'[unknown]'
    b'ping' [3760920]
    1
```


Process P1
'high' addr
main() stack [rw-]
Library mappings
Heap
Uninitialized
Initialized
Text [r-x]
main
0x0
Userspace virtual addresses
Process P2
main() stack [rw-]
Library mappings
Heap
Uninitialized
Initialized
Text [r-x]
main
thrd2
thrd3
...
Process Pn
main() stack [rw-]
Library mappings
Heap
Uninitialized
Initialized
Text [r-x]
main
thrd2
Legend
main() thread
app thread
usermode app thread stack
User Space
Kernel Space / kernel segment
P1
main
task_struct
Kernel stack
P2
main
task_struct
Kernel stack
thrd2
task_struct
Kernel stack
thrd3
task_struct
Kernel stack
...
Pn
main
task_struct
Kernel stack
thrd2
task_struct
Kernel stack
Kernel
[(user and kernel) thread creation/destruction, CPU scheduling, synchronization primitives, signaling, timers, interrupt handling, namespaces, cgroups, module support, etc
Memory management
VFS
Networking
Virt
sound
IPC
Block IO
task_struct
kthrd1
Kernel stack
task_struct
kthrd2
Kernel stack
task_struct
kthrdn
Kernel stack
init
security
arch-specific
Device Drivers
Network (NIC)
Character
Storage
Hardware Peripherals
CPU(s)
MMU / RAM

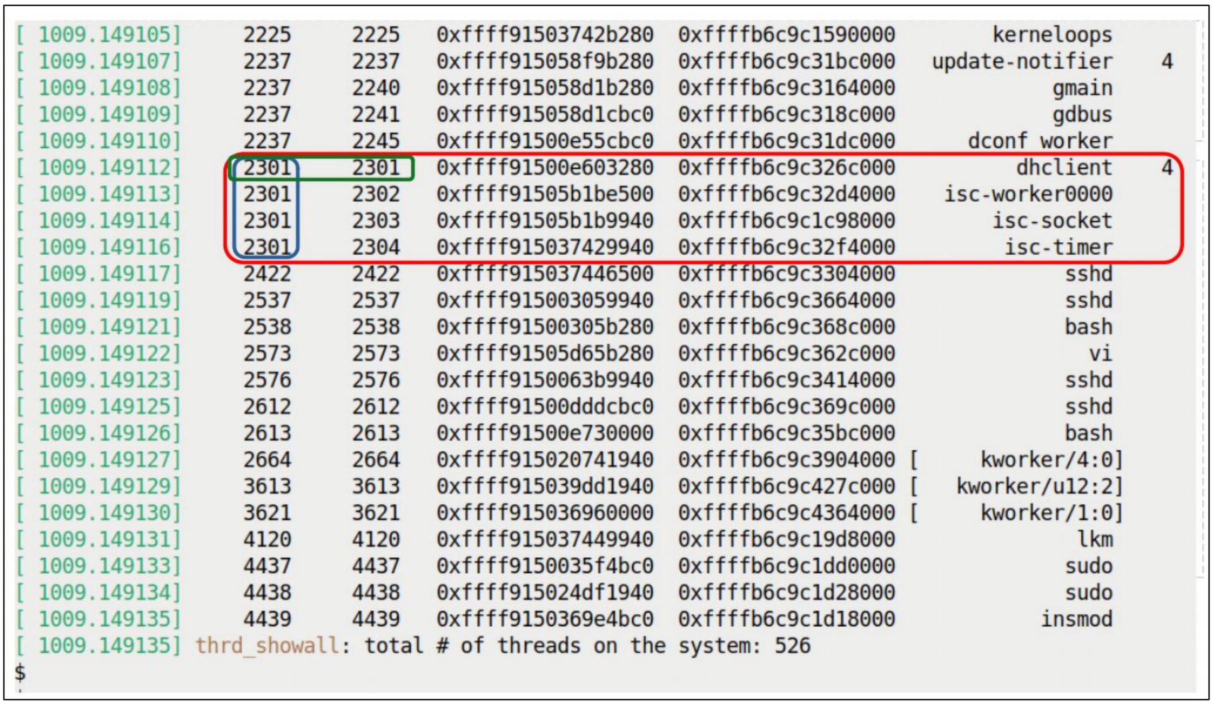

The task structure
Inherited across fork()
OFDT (open files)
VFS data
Hardware Context
Credentials / Caps
Paging Tables (PTEs)
Signal Handling
Sched info
Resource Limits
Namespaces
Thread TLS
Profiling info
Security - LSMs - seccomp
IPC Structures
Pending/blocked signals
PID, PPID
AIO
Not inherited across fork()
Locks
Timers, Alarms
Audit Info
semop adj



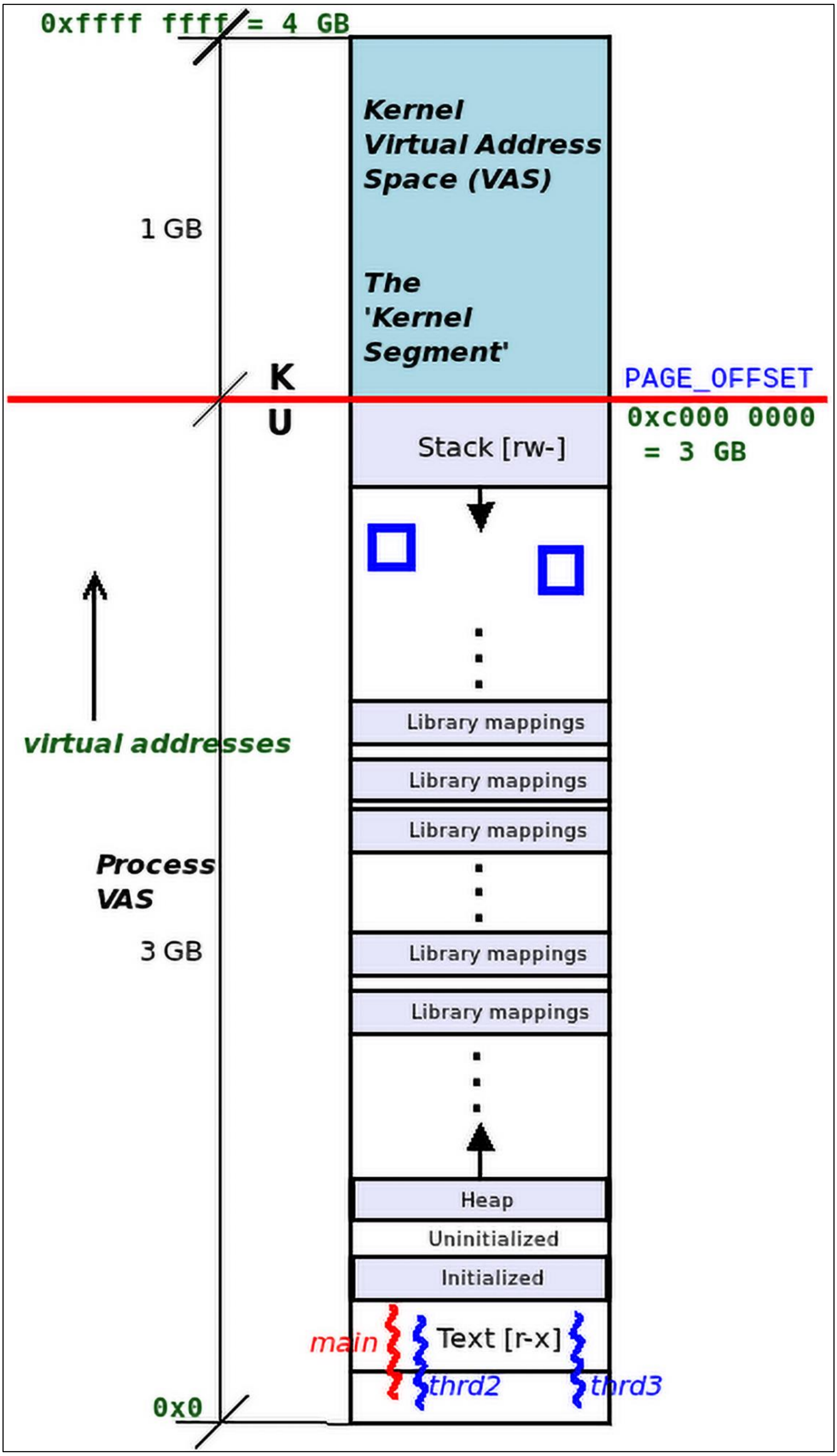
```
$ sudo dmesg -C
$ sudo insmod ./current_affairs.ko ; lsmod|grep current_affairs
current_affairs        16384  0
$ sleep 1
$ sudo rmmod current_affairs
$ sudo dmesg
[  295.072202] current_affairs:current_affairs_init(): inserted
[  295.072208] current_affairs:current_affairs_init(): sizeof(struct task_struct)=13120
[  295.072212] current_affairs:show_ctx():
[  295.072215] current_affairs:show_ctx(): we're running in process context ::
                name        : insmod
                PID         :   3303
                TGID        :   3303
                UID         :      0
                EUID        :      0 (have root)
                state       : R
                current (ptr to our process context's task_struct) :
                              0xffff88804d7c0000 (0xffff88804d7c0000)
                stack start : 0xffffc90003048000 (0xffffc90003048000)
[  300.789069] current_affairs:show_ctx():
[  300.789076] current_affairs:show_ctx(): we're running in process context ::
                name        : rmmod
                PID         :   3312
                TGID        :   3312
                UID         :      0
                EUID        :      0 (have root)
                state       : R
                current (ptr to our process context's task_struct) :
                              0xffff88801ce6a780 (0xffff88801ce6a780)
                stack start : 0xffffc90002768000 (0xffffc90002768000)
[  300.789085] current_affairs:current_affairs_exit(): removed
$
```

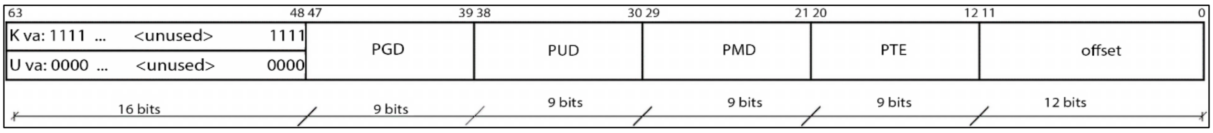


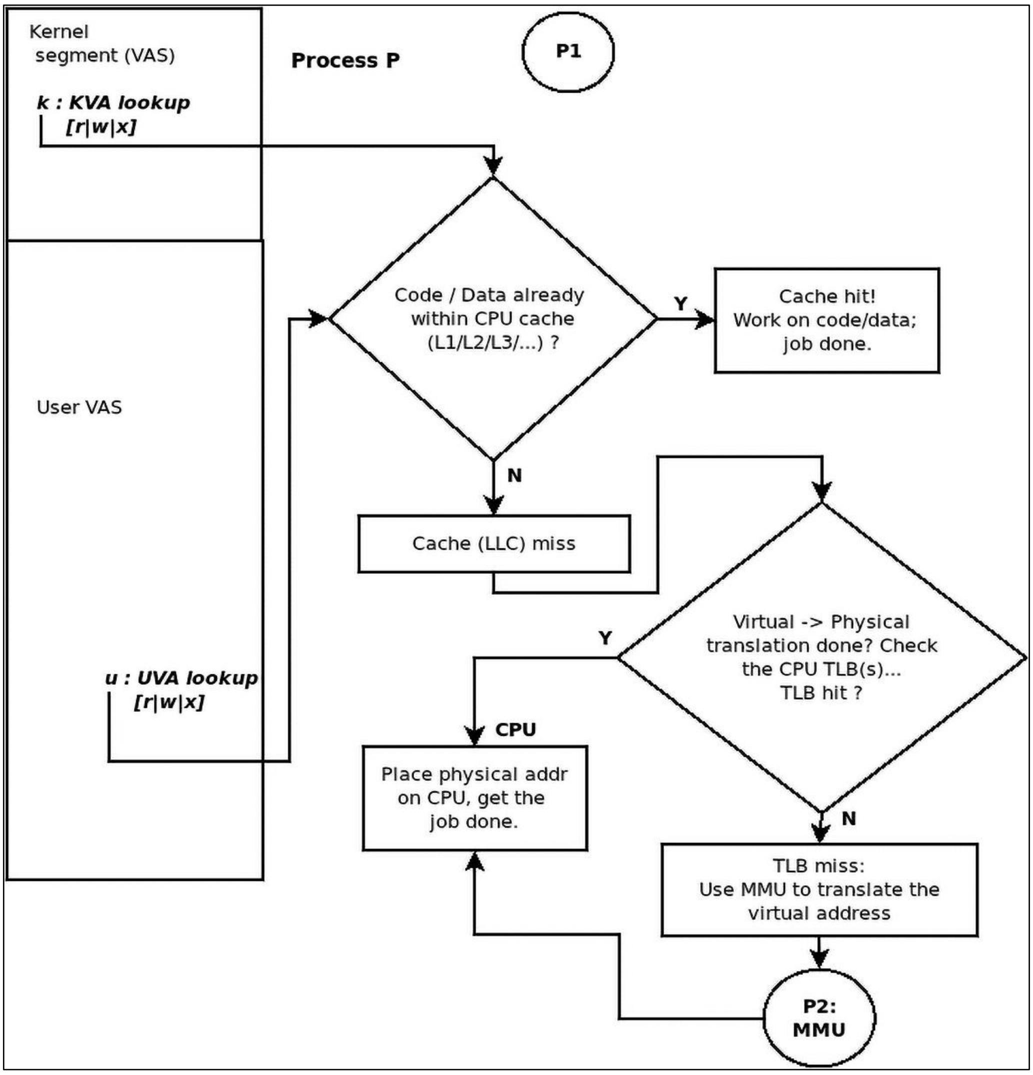
```
[ 1685.208236] prcs_showall: inserted
[ 1685.208239] prcs_showall:      Name          |  TGID   |    PID   |  RUID |  EUID
[ 1685.208241] prcs_showall: systemd            |        1|         1|      0|      0
[ 1685.208242] prcs_showall: kthreadd           |        2|         2|      0|      0
[ 1685.208243] prcs_showall: rcu_gp             |        3|         3|      0|      0
[ 1685.208244] prcs_showall: rcu_par_gp         |        4|         4|      0|      0
[ 1685.208245] prcs_showall: slub_flushwq       |        5|         5|      0|      0
[ 1685.208246] prcs_showall: netns              |        6|         6|      0|      0
[ 1685.208247] prcs_showall: kworker/0:0H       |        8|         8|      0|      0
[ 1685.208248] prcs_showall: mm_percpu_wq       |       10|        10|      0|      0
[ 1685.208249] prcs_showall: rcu_tasks_kthre    |       11|        11|      0|      0
[ 1685.208250] prcs_showall: rcu_tasks_rude_    |       12|        12|      0|      0
[ 1685.208251] prcs_showall: rcu_tasks_trace    |       13|        13|      0|      0
[ 1685.208252] prcs_showall: ksoftirqd/0        |       14|        14|      0|      0
[ 1685.208253] prcs_showall: rcu_preempt        |       15|        15|      0|      0
[ 1685.208254] prcs_showall: migration/0        |       16|        16|      0|      0
```

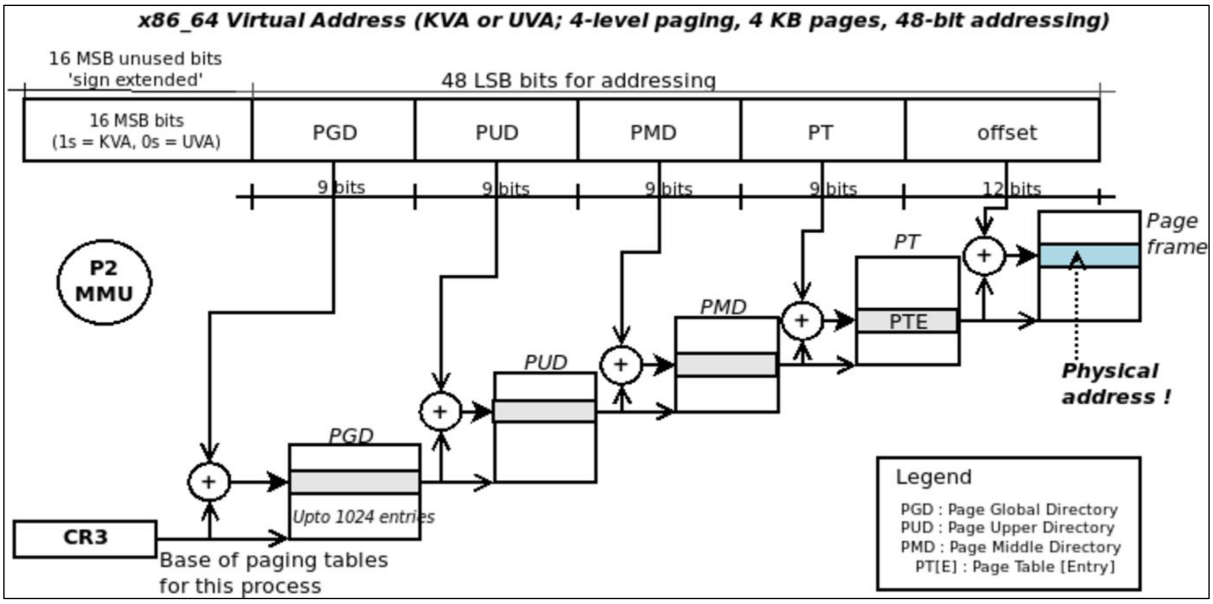


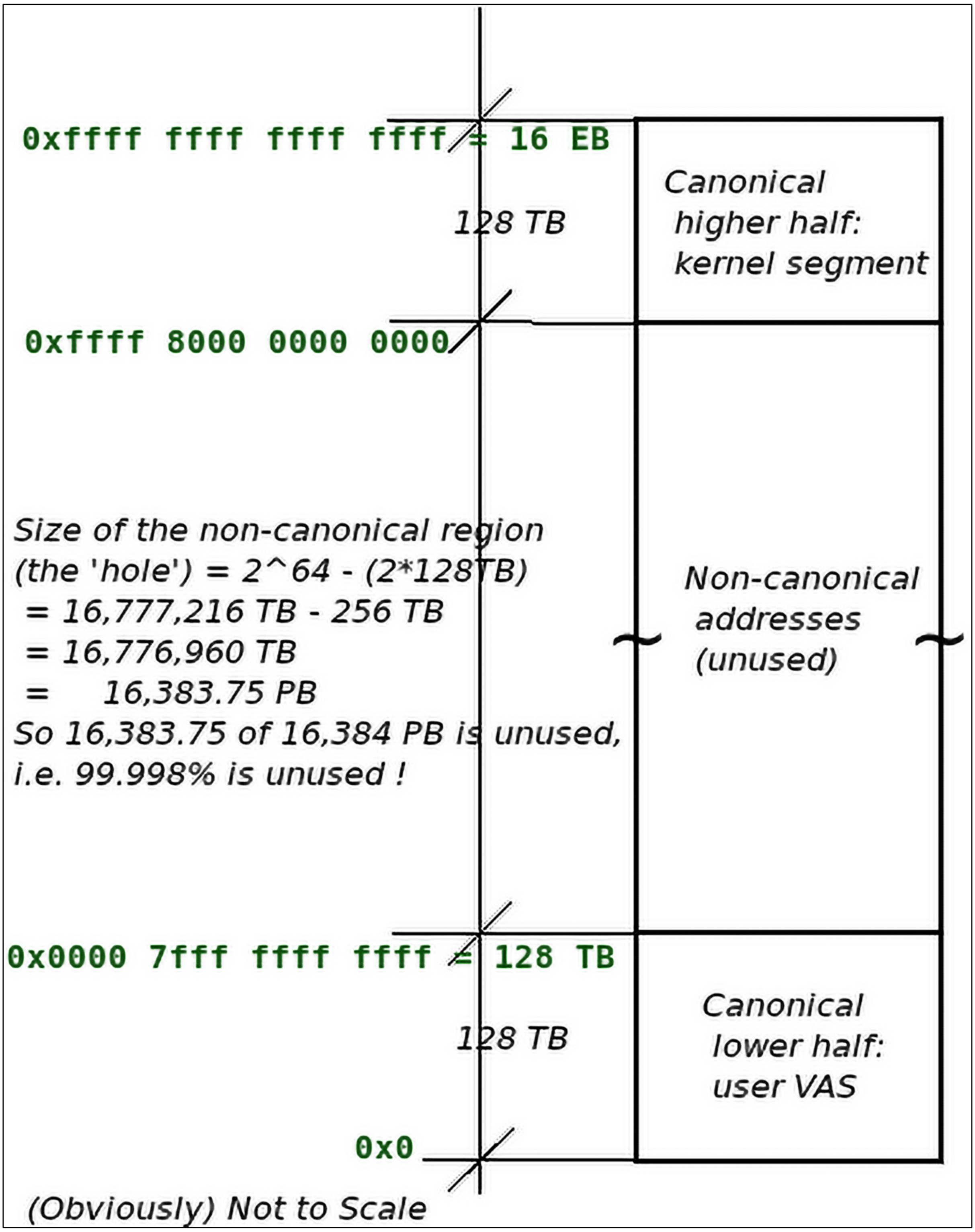
```
[ 1009.149105]      2225     2225   0xffff91503742b280  0xffffb6c9c1590000        kerneloops
[ 1009.149107]      2237     2237   0xffff915058f9b280  0xffffb6c9c31bc000   update-notifier    4
[ 1009.149108]      2237     2240   0xffff915058d1b280  0xffffb6c9c3164000             gmain
[ 1009.149109]      2237     2241   0xffff915058d1cbc0  0xffffb6c9c318c000             gdbus
[ 1009.149110]      2237     2245   0xffff91500e55cbc0  0xffffb6c9c31dc000      dconf worker
[ 1009.149112]      2301     2301   0xffff91500e603280  0xffffb6c9c326c000          dhclient    4
[ 1009.149113]      2301     2302   0xffff91505b1be500  0xffffb6c9c32d4000    isc-worker0000
[ 1009.149114]      2301     2303   0xffff91505b1b9940  0xffffb6c9c1c98000        isc-socket
[ 1009.149116]      2301     2304   0xffff915037429940  0xffffb6c9c32f4000         isc-timer
[ 1009.149117]      2422     2422   0xffff915037446500  0xffffb6c9c3304000              sshd
[ 1009.149119]      2537     2537   0xffff915003059940  0xffffb6c9c3664000              sshd
[ 1009.149121]      2538     2538   0xffff91500305b280  0xffffb6c9c368c000              bash
[ 1009.149122]      2573     2573   0xffff91505d65b280  0xffffb6c9c362c000                vi
[ 1009.149123]      2576     2576   0xffff9150063b9940  0xffffb6c9c3414000              sshd
[ 1009.149125]      2612     2612   0xffff91500dddcbc0  0xffffb6c9c369c000              sshd
[ 1009.149126]      2613     2613   0xffff91500e730000  0xffffb6c9c35bc000              bash
[ 1009.149127]      2664     2664   0xffff915020741940  0xffffb6c9c3904000 [     kworker/4:0]
[ 1009.149129]      3613     3613   0xffff915039dd1940  0xffffb6c9c427c000 [   kworker/u12:2]
[ 1009.149130]      3621     3621   0xffff915036960000  0xffffb6c9c4364000 [     kworker/1:0]
[ 1009.149131]      4120     4120   0xffff915037449940  0xffffb6c9c19d8000               lkm
[ 1009.149133]      4437     4437   0xffff9150035f4bc0  0xffffb6c9c1dd0000              sudo
[ 1009.149134]      4438     4438   0xffff915024df1940  0xffffb6c9c1d28000              sudo
[ 1009.149135]      4439     4439   0xffff9150369e4bc0  0xffffb6c9c1d18000            insmod
[ 1009.149135] thrd_showall: total # of threads on the system: 526
$
```

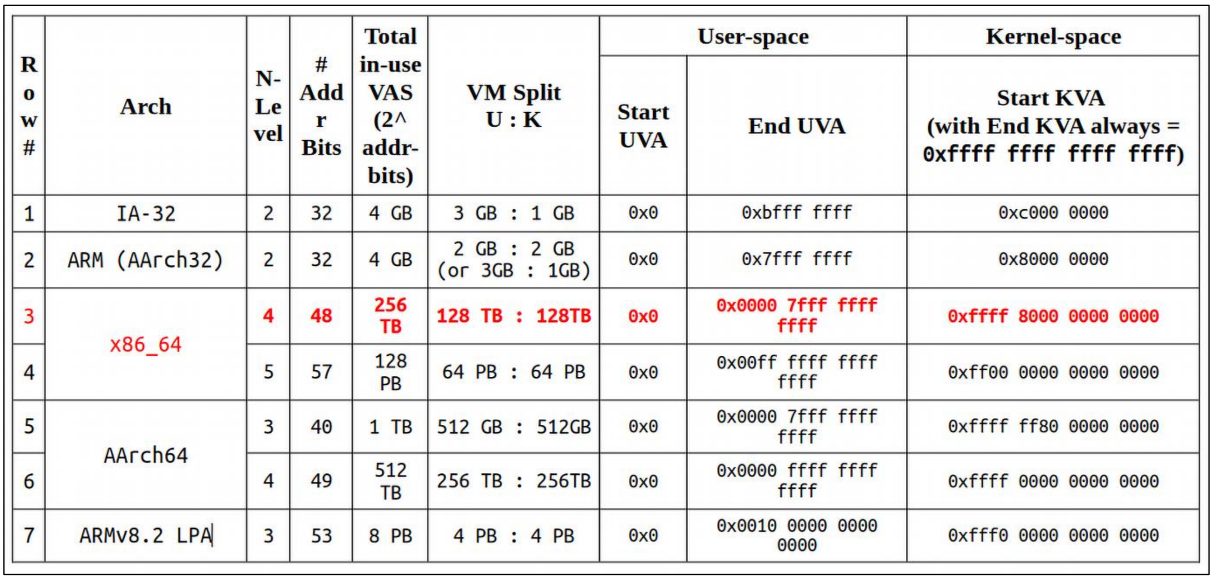

0xffff ffff = 4 GB
Kernel Virtual Address Space (VAS)
1 GB
The 'Kernel Segment'
K
U
PAGE_OFFSET
0xc000 0000 = 3 GB
Stack [rw-]
virtual addresses
Library mappings
Library mappings
Library mappings
Process VAS
3 GB
Library mappings
Library mappings
Heap
Uninitialized
Initialized
main
Text [r-x]
thrd2
thrd3
0x0


63
48 47
39 38
30 29
21 20
12 11
0
K va: 1111 ... <unused> 1111
U va: 0000 ... <unused> 0000
PGD
PUD
PMD
PTE
offset
16 bits
9 bits
9 bits
9 bits
9 bits
12 bits

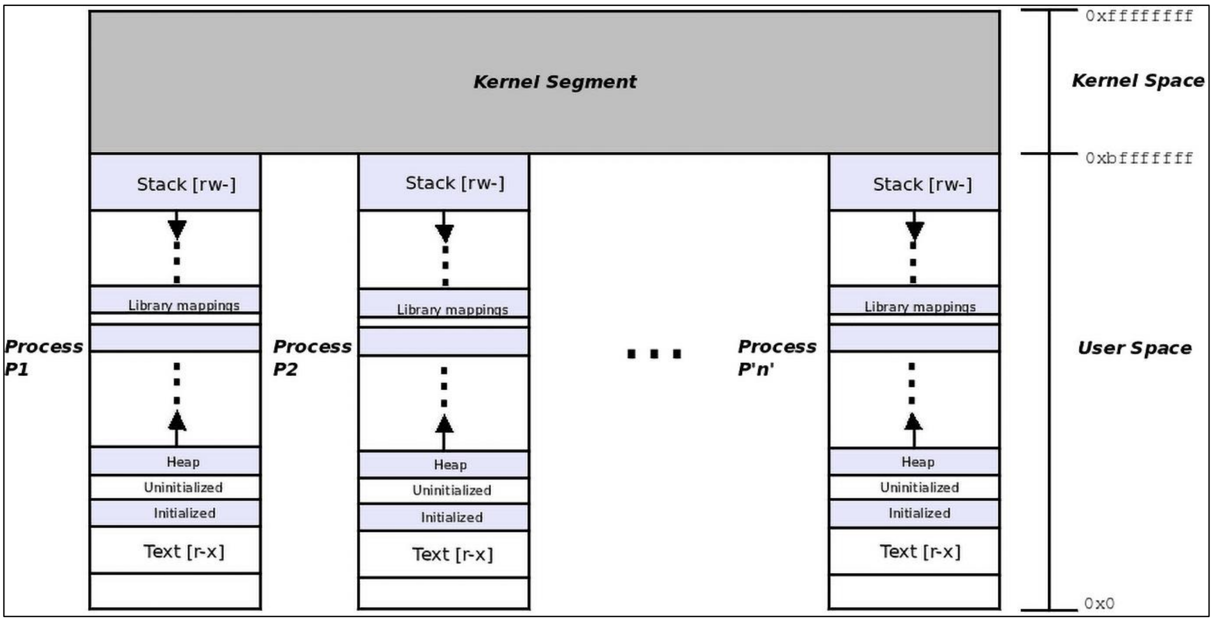

Kernel
segment (VAS)
k : KVA lookup
[r|w|x]
User VAS
u : UVA lookup
[r|w|x]
Process P
P1
Code / Data already
within CPU cache
(L1/L2/L3/...) ?
Y
Cache hit!
Work on code/data;
job done.
N
Cache (LLC) miss
Virtual -> Physical
translation done? Check
the CPU TLB(s)...
TLB hit ?
Y
CPU
Place physical addr
on CPU, get the
job done.
N
TLB miss:
Use MMU to translate the
virtual address
P2:
MMU

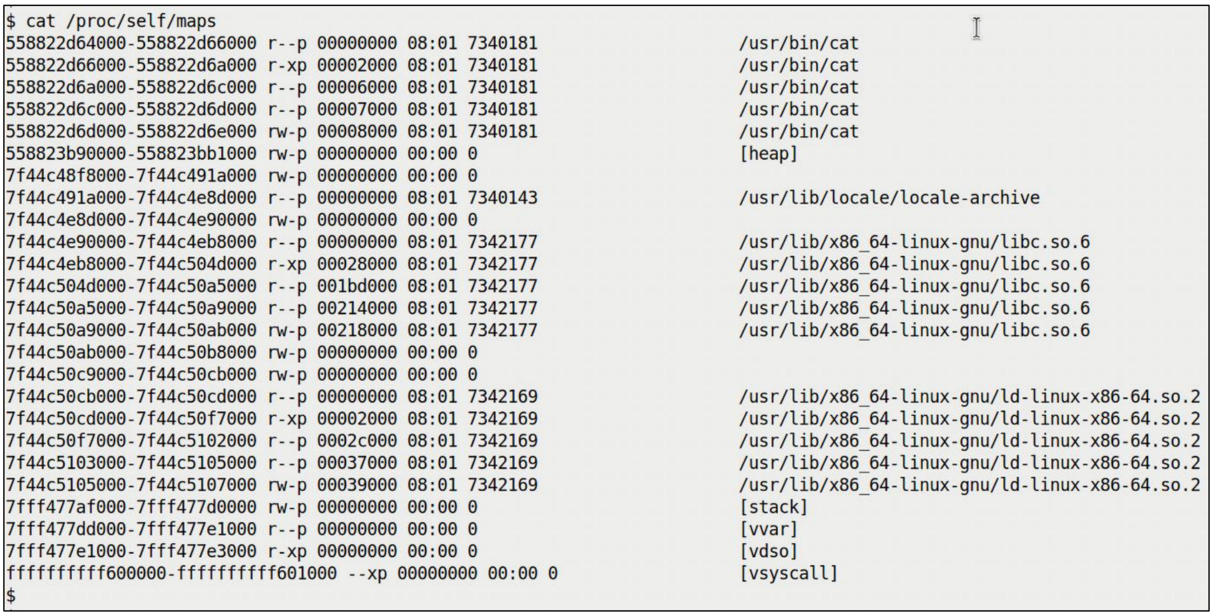

x86_64 Virtual Address (KVA or UVA; 4-level paging, 4 KB pages, 48-bit addressing)
16 MSB unused bits 'sign extended'
48 LSB bits for addressing
16 MSB bits (1s = KVA, 0s = UVA)
PGD
PUD
PMD
PT
offset
9 bits
9 bits
9 bits
9 bits
12 bits
P2 MMU
PT
PTE
Page frame
PMD
PUD
PGD
Physical address !
Upto 1024 entries
CR3
Base of paging tables for this process
Legend
PGD : Page Global Directory
PUD : Page Upper Directory
PMD : Page Middle Directory
PT[E] : Page Table [Entry]


0xffff ffff ffff ffff = 16 EB
128 TB
Canonical higher half: kernel segment
0xffff 8000 0000 0000
Size of the non-canonical region (the 'hole') = 2^64 - (2*128TB)
= 16,777,216 TB - 256 TB
= 16,776,960 TB
= 16,383.75 PB
So 16,383.75 of 16,384 PB is unused,
i.e. 99.998% is unused !
Non-canonical addresses (unused)
0x0000 7fff ffff ffff = 128 TB
128 TB
Canonical lower half: user VAS
0x0
(Obviously) Not to Scale

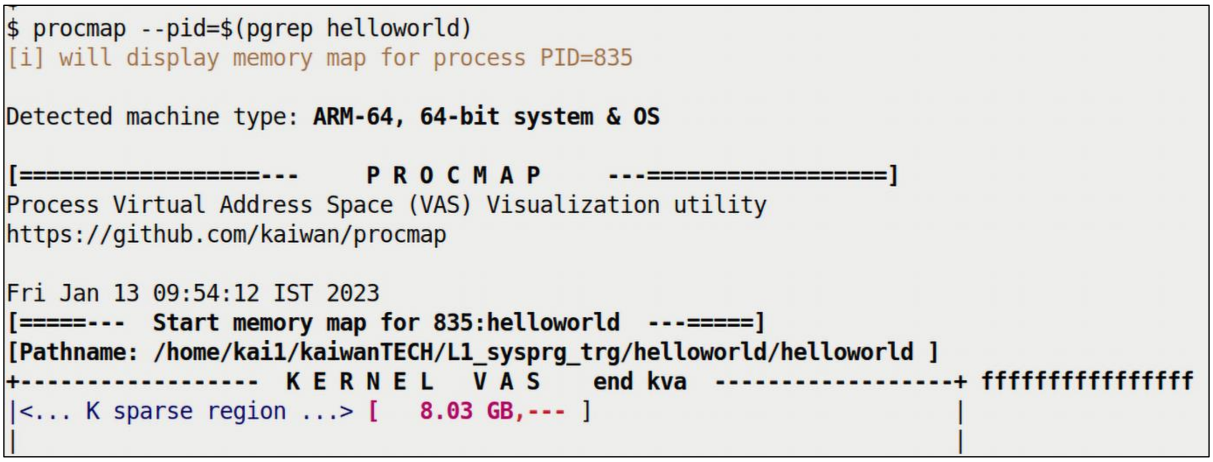


Row #	Arch	N-Level	# Addr Bits	Total in-use VAS (2^ addr-bits)	VM Split U : K	User-space		Kernel-space
						Start UVA	End UVA	Start KVA (with End KVA always = `0xffff ffff ffff ffff`)
1	IA-32	2	32	4 GB	3 GB : 1 GB	0x0	0xbfff ffff	0xc000 0000
2	ARM (AArch32)	2	32	4 GB	2 GB : 2 GB (or 3GB : 1GB)	0x0	0x7fff ffff	0x8000 0000
3	x86_64	4	48	256 TB	128 TB : 128TB	0x0	0x0000 7fff ffff ffff	0xffff 8000 0000 0000
4		5	57	128 PB	64 PB : 64 PB	0x0	0x00ff ffff ffff ffff	0xff00 0000 0000 0000
5	AArch64	3	40	1 TB	512 GB : 512GB	0x0	0x0000 7fff ffff ffff	0xffff ff80 0000 0000
6		4	49	512 TB	256 TB : 256TB	0x0	0x0000 ffff ffff ffff	0xffff 0000 0000 0000
7	ARMv8.2 LPA	3	53	8 PB	4 PB : 4 PB	0x0	0x0010 0000 0000 0000	0xfff0 0000 0000 0000

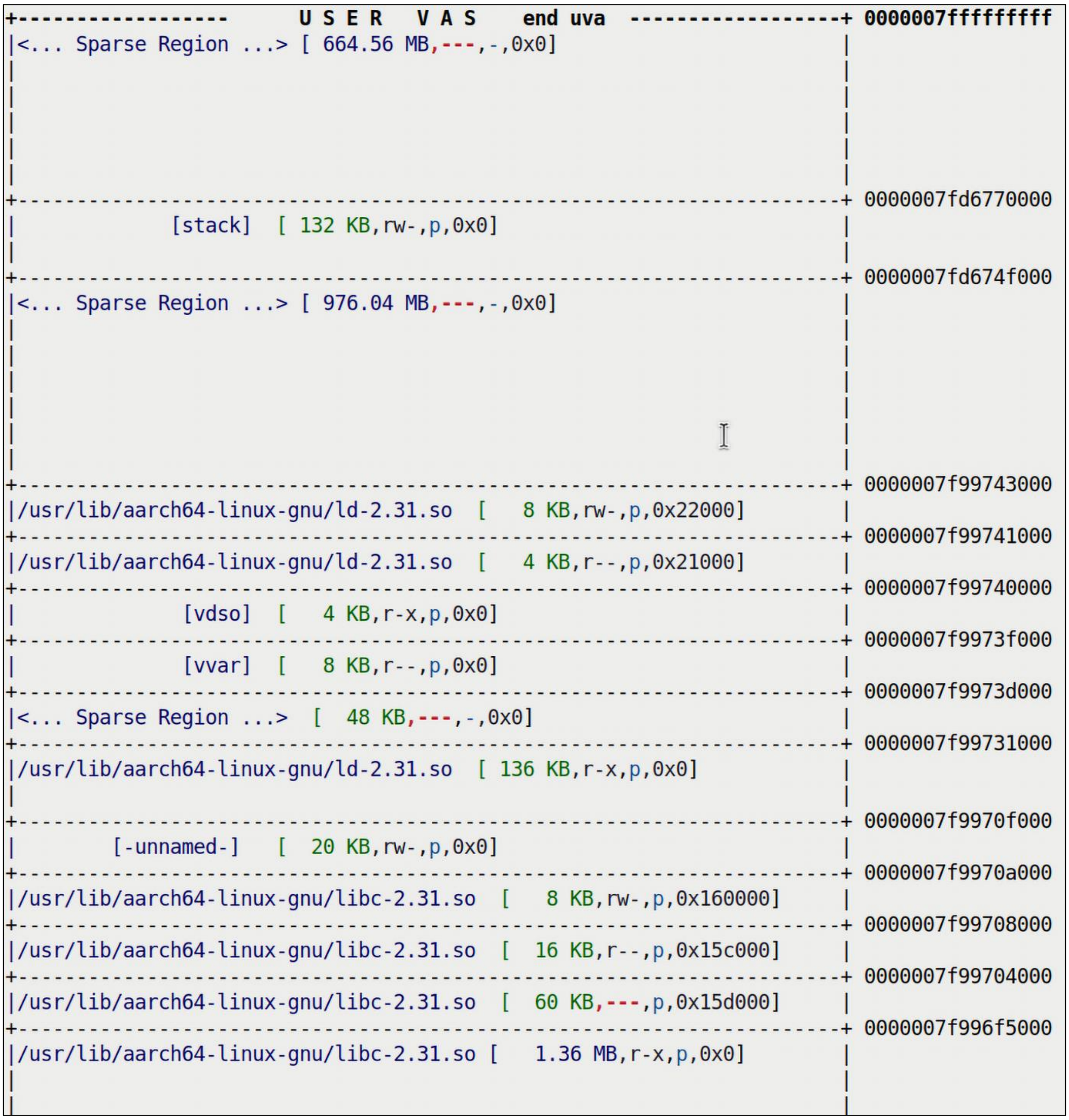




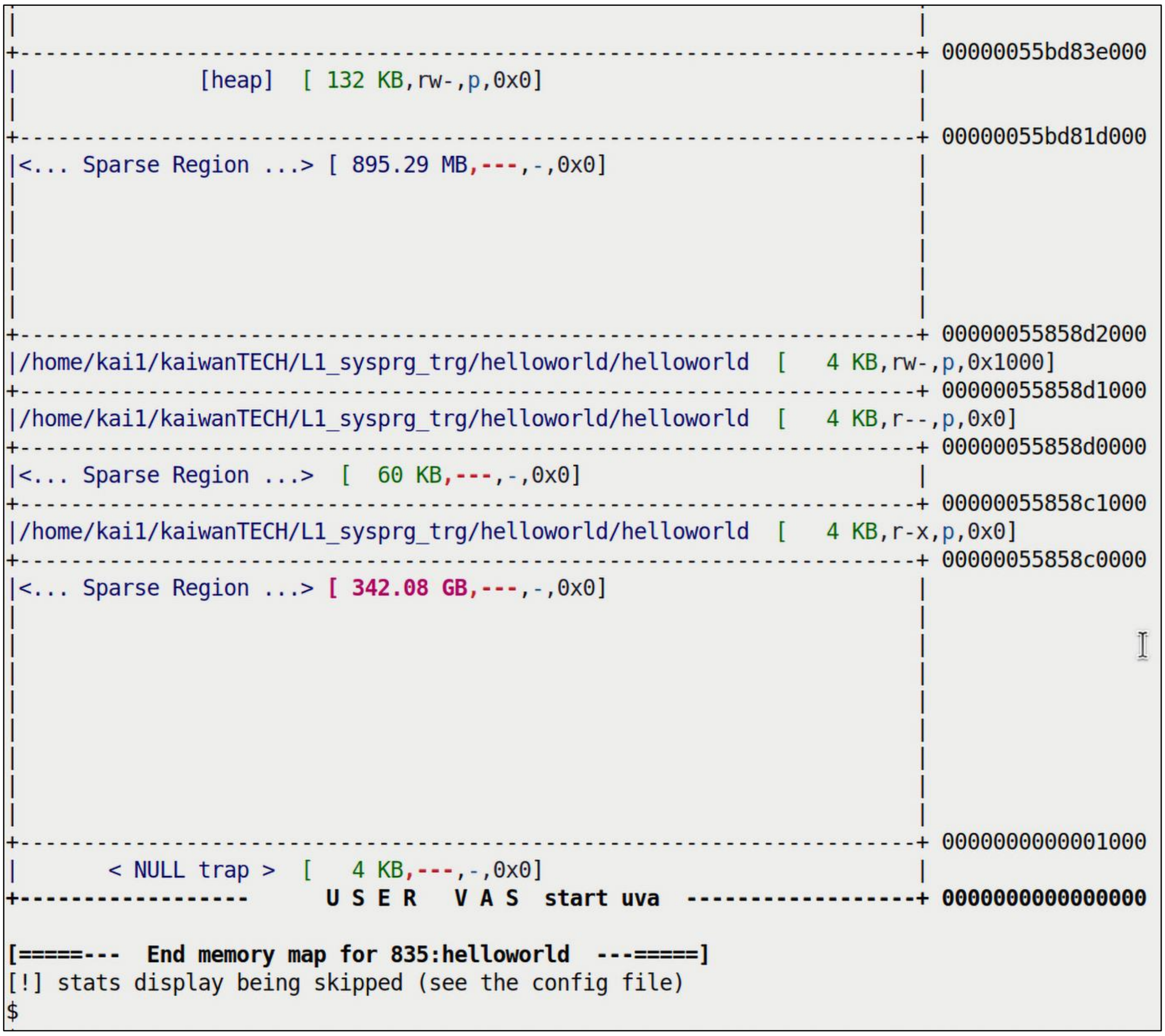
```
$ cat /proc/self/maps
558822d64000-558822d66000 r--p 00000000 08:01 7340181                    /usr/bin/cat
558822d66000-558822d6a000 r-xp 00002000 08:01 7340181                    /usr/bin/cat
558822d6a000-558822d6c000 r--p 00006000 08:01 7340181                    /usr/bin/cat
558822d6c000-558822d6d000 r--p 00007000 08:01 7340181                    /usr/bin/cat
558822d6d000-558822d6e000 rw-p 00008000 08:01 7340181                    /usr/bin/cat
558823b90000-558823bb1000 rw-p 00000000 00:00 0                          [heap]
7f44c48f8000-7f44c491a000 rw-p 00000000 00:00 0
7f44c491a000-7f44c4e8d000 r--p 00000000 08:01 7340143                    /usr/lib/locale/locale-archive
7f44c4e8d000-7f44c4e90000 rw-p 00000000 00:00 0
7f44c4e90000-7f44c4eb8000 r--p 00000000 08:01 7342177                    /usr/lib/x86_64-linux-gnu/libc.so.6
7f44c4eb8000-7f44c504d000 r-xp 00028000 08:01 7342177                    /usr/lib/x86_64-linux-gnu/libc.so.6
7f44c504d000-7f44c50a5000 r--p 001bd000 08:01 7342177                    /usr/lib/x86_64-linux-gnu/libc.so.6
7f44c50a5000-7f44c50a9000 r--p 00214000 08:01 7342177                    /usr/lib/x86_64-linux-gnu/libc.so.6
7f44c50a9000-7f44c50ab000 rw-p 00218000 08:01 7342177                    /usr/lib/x86_64-linux-gnu/libc.so.6
7f44c50ab000-7f44c50b8000 rw-p 00000000 00:00 0
7f44c50c9000-7f44c50cb000 rw-p 00000000 00:00 0
7f44c50cb000-7f44c50cd000 r--p 00000000 08:01 7342169                    /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7f44c50cd000-7f44c50f7000 r-xp 00002000 08:01 7342169                    /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7f44c50f7000-7f44c5102000 r--p 0002c000 08:01 7342169                    /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7f44c5103000-7f44c5105000 r--p 00037000 08:01 7342169                    /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7f44c5105000-7f44c5107000 rw-p 00039000 08:01 7342169                    /usr/lib/x86_64-linux-gnu/ld-linux-x86-64.so.2
7fff477af000-7fff477d0000 rw-p 00000000 00:00 0                          [stack]
7fff477dd000-7fff477e1000 r--p 00000000 00:00 0                          [vvar]
7fff477e1000-7fff477e3000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 --xp 00000000 00:00 0                  [vsyscall]
$
```



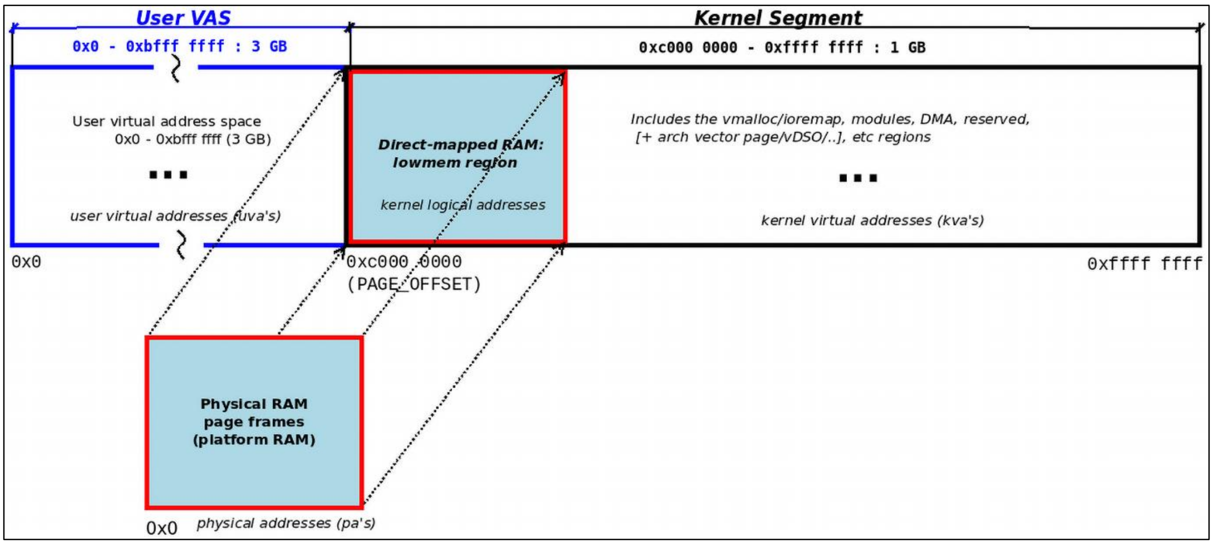
```
$ procmap --pid=$(pgrep helloworld)
[i] will display memory map for process PID=835

Detected machine type: ARM-64, 64-bit system & OS

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

Fri Jan 13 09:54:12 IST 2023
[=====---  Start memory map for 835:helloworld  ---=====]
[Pathname: /home/kai1/kaiwanTECH/L1_sysprg_trg/helloworld/helloworld ]
+------------------  K E R N E L   V A S    end kva  ------------------+ ffffffffffffffff
|<... K sparse region ...> [   8.03 GB,--- ]                           |
|                                                                      |
```

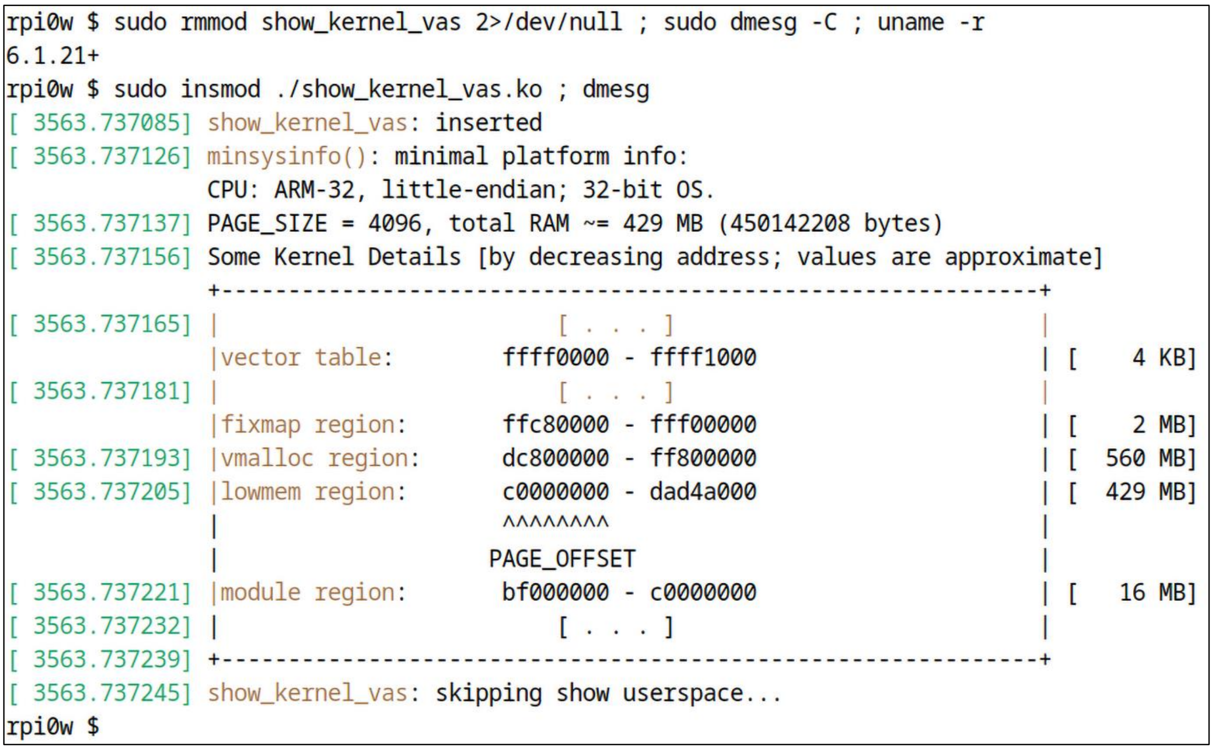


```
+------------------      U S E R   V A S    end uva  -----------------+ 0000007fffffffff
|<... Sparse Region ...> [ 664.56 MB,---,-,0x0]                        |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
+----------------------------------------------------------------------+ 0000007fd6770000
|             [stack]  [ 132 KB,rw-,p,0x0]                             |
|                                                                      |
+----------------------------------------------------------------------+ 0000007fd674f000
|<... Sparse Region ...> [ 976.04 MB,---,-,0x0]                        |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
+----------------------------------------------------------------------+ 0000007f99743000
|/usr/lib/aarch64-linux-gnu/ld-2.31.so  [   8 KB,rw-,p,0x22000]        |
+----------------------------------------------------------------------+ 0000007f99741000
|/usr/lib/aarch64-linux-gnu/ld-2.31.so  [   4 KB,r--,p,0x21000]        |
+----------------------------------------------------------------------+ 0000007f99740000
|              [vdso]  [   4 KB,r-x,p,0x0]                             |
+----------------------------------------------------------------------+ 0000007f9973f000
|              [vvar]  [   8 KB,r--,p,0x0]                             |
+----------------------------------------------------------------------+ 0000007f9973d000
|<... Sparse Region ...>  [  48 KB,---,-,0x0]                          |
+----------------------------------------------------------------------+ 0000007f99731000
|/usr/lib/aarch64-linux-gnu/ld-2.31.so  [ 136 KB,r-x,p,0x0]            |
|                                                                      |
+----------------------------------------------------------------------+ 0000007f9970f000
|        [-unnamed-]   [  20 KB,rw-,p,0x0]                             |
+----------------------------------------------------------------------+ 0000007f9970a000
|/usr/lib/aarch64-linux-gnu/libc-2.31.so  [   8 KB,rw-,p,0x160000]     |
+----------------------------------------------------------------------+ 0000007f99708000
|/usr/lib/aarch64-linux-gnu/libc-2.31.so  [  16 KB,r--,p,0x15c000]     |
+----------------------------------------------------------------------+ 0000007f99704000
|/usr/lib/aarch64-linux-gnu/libc-2.31.so  [  60 KB,---,p,0x15d000]     |
+----------------------------------------------------------------------+ 0000007f996f5000
|/usr/lib/aarch64-linux-gnu/libc-2.31.so [   1.36 MB,r-x,p,0x0]        |
|                                                                      |
|                                                                      |
```

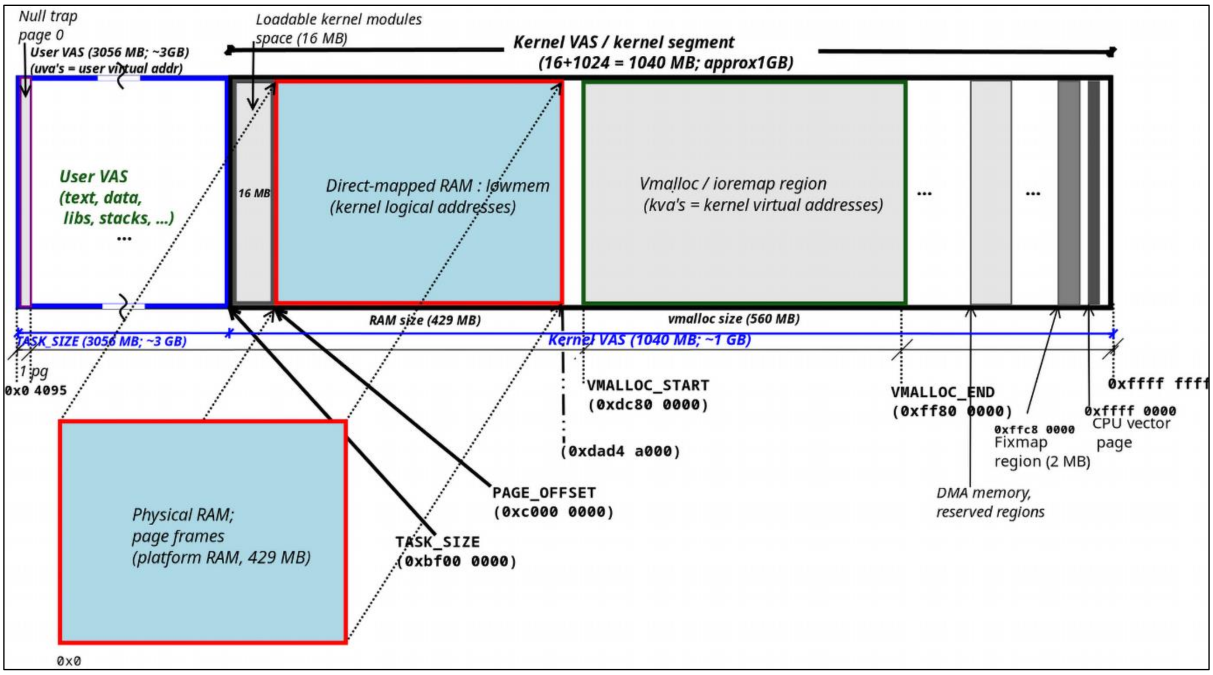


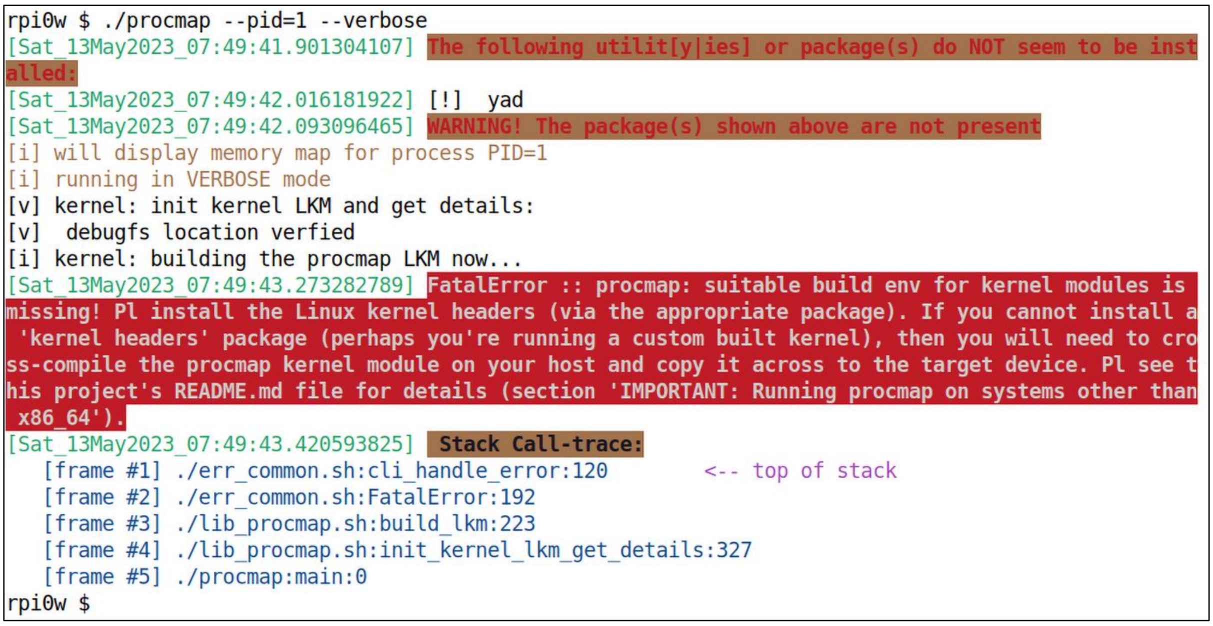
```
|                                                                       |
+-----------------------------------------------------------------------+ 00000055bd83e000
|              [heap]  [ 132 KB,rw-,p,0x0]                              |
|                                                                       |
+-----------------------------------------------------------------------+ 00000055bd81d000
|<... Sparse Region ...> [ 895.29 MB,---,-,0x0]                         |
|                                                                       |
|                                                                       |
|                                                                       |
|                                                                       |
|                                                                       |
+-----------------------------------------------------------------------+ 00000055858d2000
|/home/kai1/kaiwanTECH/L1_sysprg_trg/helloworld/helloworld  [   4 KB,rw-,p,0x1000]
+-----------------------------------------------------------------------+ 00000055858d1000
|/home/kai1/kaiwanTECH/L1_sysprg_trg/helloworld/helloworld  [   4 KB,r--,p,0x0]
+-----------------------------------------------------------------------+ 00000055858d0000
|<... Sparse Region ...>  [  60 KB,---,-,0x0]                           |
+-----------------------------------------------------------------------+ 00000055858c1000
|/home/kai1/kaiwanTECH/L1_sysprg_trg/helloworld/helloworld  [   4 KB,r-x,p,0x0]
+-----------------------------------------------------------------------+ 00000055858c0000
|<... Sparse Region ...> [ 342.08 GB,---,-,0x0]                         |
|                                                                       |
|                                                                       |
|                                                                       |
|                                                                       |
|                                                                       |
|                                                                       |
|                                                                       |
|                                                                       |
+-----------------------------------------------------------------------+ 0000000000001000
|       < NULL trap >  [   4 KB,---,-,0x0]                              |
+------------------      U S E R   V A S  start uva  ------------------+ 0000000000000000

[=====---  End memory map for 835:helloworld  ---=====]
[!] stats display being skipped (see the config file)
$
```


User VAS
0x0 - 0xbfff ffff : 3 GB
Kernel Segment
0xc000 0000 - 0xffff ffff : 1 GB
User virtual address space
0x0 - 0xbfff ffff (3 GB)
...
user virtual addresses (uva's)
Direct-mapped RAM:
lowmem region
kernel logical addresses
Includes the vmalloc/ioremap, modules, DMA, reserved,
[+ arch vector page/vDSO/..], etc regions
...
kernel virtual addresses (kva's)
0x0
0xc000 0000
(PAGE_OFFSET)
0xffff ffff
Physical RAM
page frames
(platform RAM)
0x0
physical addresses (pa's)

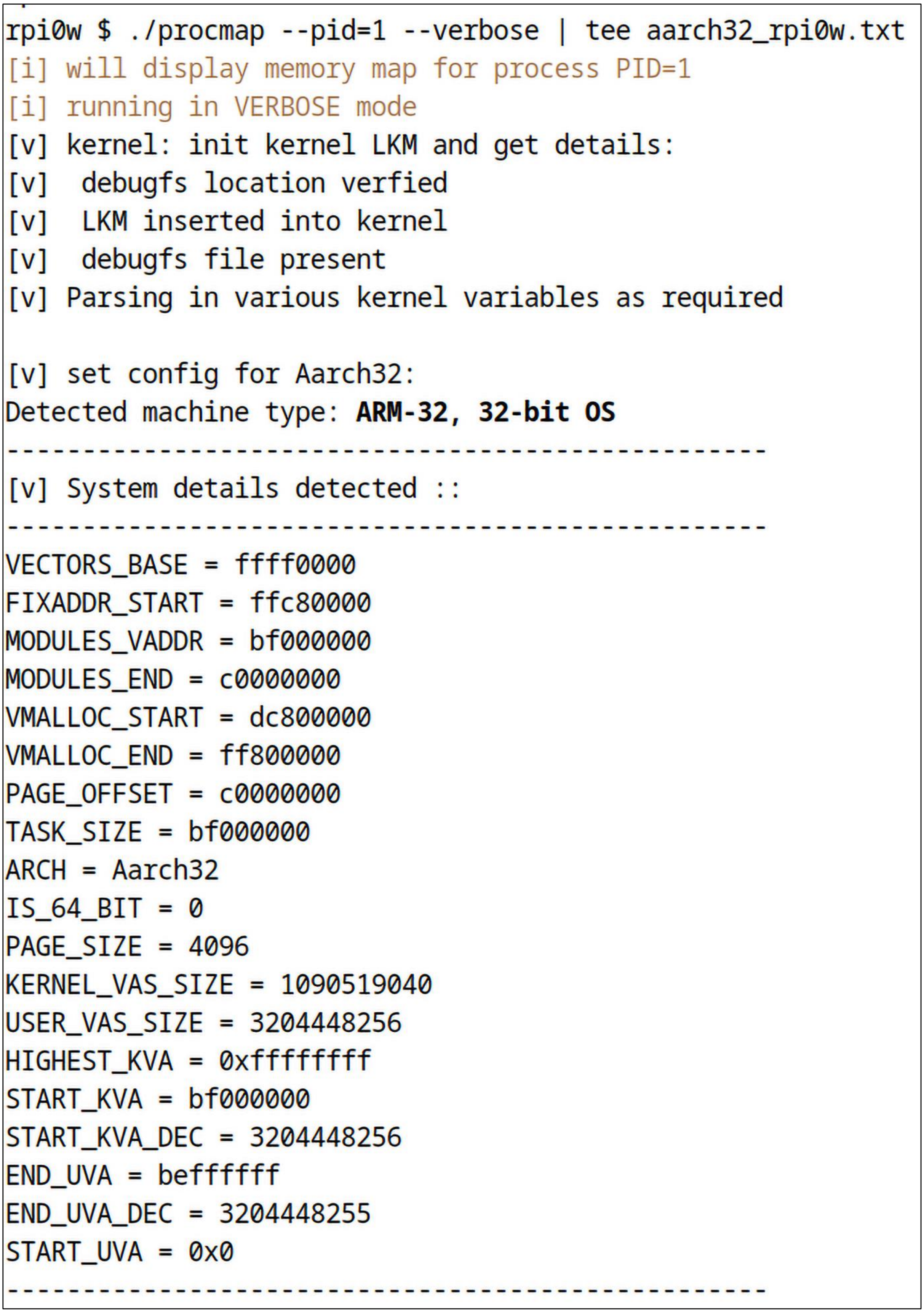


```
rpi0w $ sudo rmmod show_kernel_vas 2>/dev/null ; sudo dmesg -C ; uname -r
6.1.21+
rpi0w $ sudo insmod ./show_kernel_vas.ko ; dmesg
[ 3563.737085] show_kernel_vas: inserted
[ 3563.737126] minsysinfo(): minimal platform info:
               CPU: ARM-32, little-endian; 32-bit OS.
[ 3563.737137] PAGE_SIZE = 4096, total RAM ~= 429 MB (450142208 bytes)
[ 3563.737156] Some Kernel Details [by decreasing address; values are approximate]
               +-------------------------------------------------------------+
[ 3563.737165] |                         [ . . . ]                           |
               |vector table:        ffff0000 - ffff1000                     | [    4 KB]
[ 3563.737181] |                         [ . . . ]                           |
               |fixmap region:       ffc80000 - fff00000                     | [    2 MB]
[ 3563.737193] |vmalloc region:      dc800000 - ff800000                     | [  560 MB]
[ 3563.737205] |lowmem region:       c0000000 - dad4a000                     | [  429 MB]
               |                     ^^^^^^^^                                |
               |                    PAGE_OFFSET                              |
[ 3563.737221] |module region:       bf000000 - c0000000                     | [   16 MB]
[ 3563.737232] |                         [ . . . ]                           |
[ 3563.737239] +-------------------------------------------------------------+
[ 3563.737245] show_kernel_vas: skipping show userspace...
rpi0w $
```

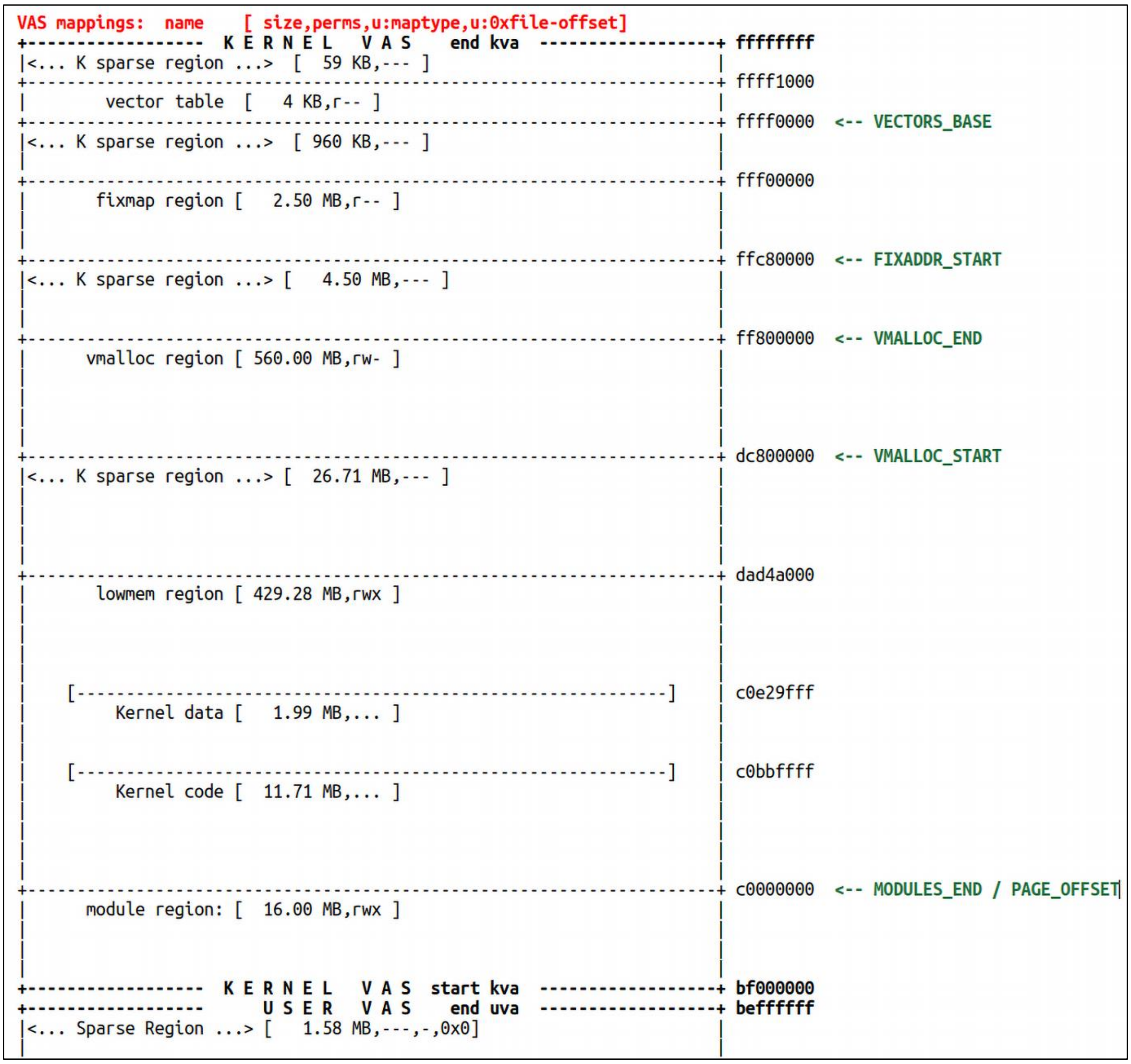




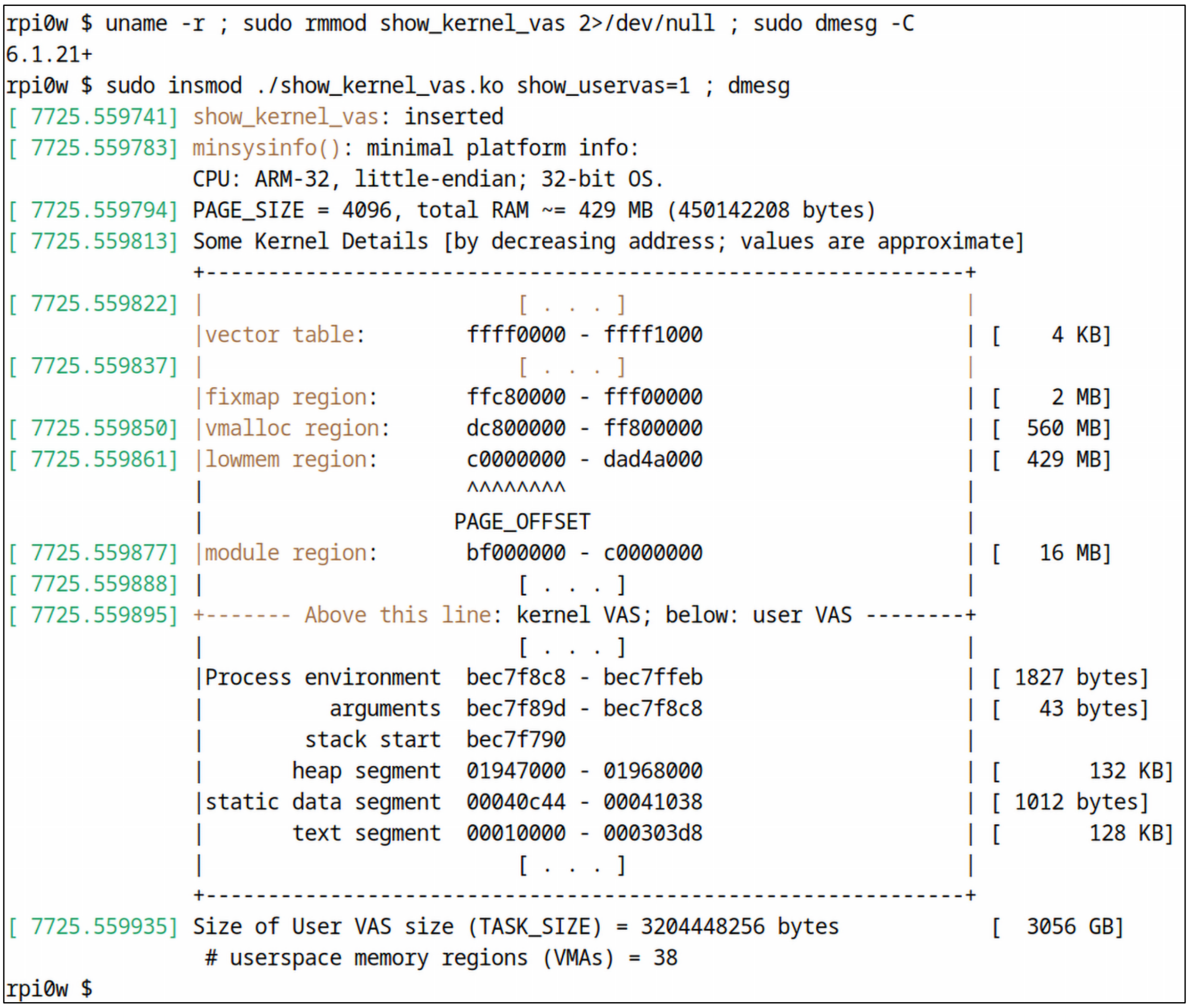
```
rpi0w $ ./procmap --pid=1 --verbose
[Sat_13May2023_07:49:41.901304107] The following utilit[y|ies] or package(s) do NOT seem to be inst
alled:
[Sat_13May2023_07:49:42.016181922] [!]  yad
[Sat_13May2023_07:49:42.093096465] WARNING! The package(s) shown above are not present
[i] will display memory map for process PID=1
[i] running in VERBOSE mode
[v] kernel: init kernel LKM and get details:
[v]  debugfs location verfied
[i] kernel: building the procmap LKM now...
[Sat_13May2023_07:49:43.273282789] FatalError :: procmap: suitable build env for kernel modules is
missing! Pl install the Linux kernel headers (via the appropriate package). If you cannot install a
 'kernel headers' package (perhaps you're running a custom built kernel), then you will need to cro
ss-compile the procmap kernel module on your host and copy it across to the target device. Pl see t
his project's README.md file for details (section 'IMPORTANT: Running procmap on systems other than
 x86_64').
[Sat_13May2023_07:49:43.420593825]  Stack Call-trace:
   [frame #1] ./err_common.sh:cli_handle_error:120        <-- top of stack
   [frame #2] ./err_common.sh:FatalError:192
   [frame #3] ./lib_procmap.sh:build_lkm:223
   [frame #4] ./lib_procmap.sh:init_kernel_lkm_get_details:327
   [frame #5] ./procmap:main:0
rpi0w $
```



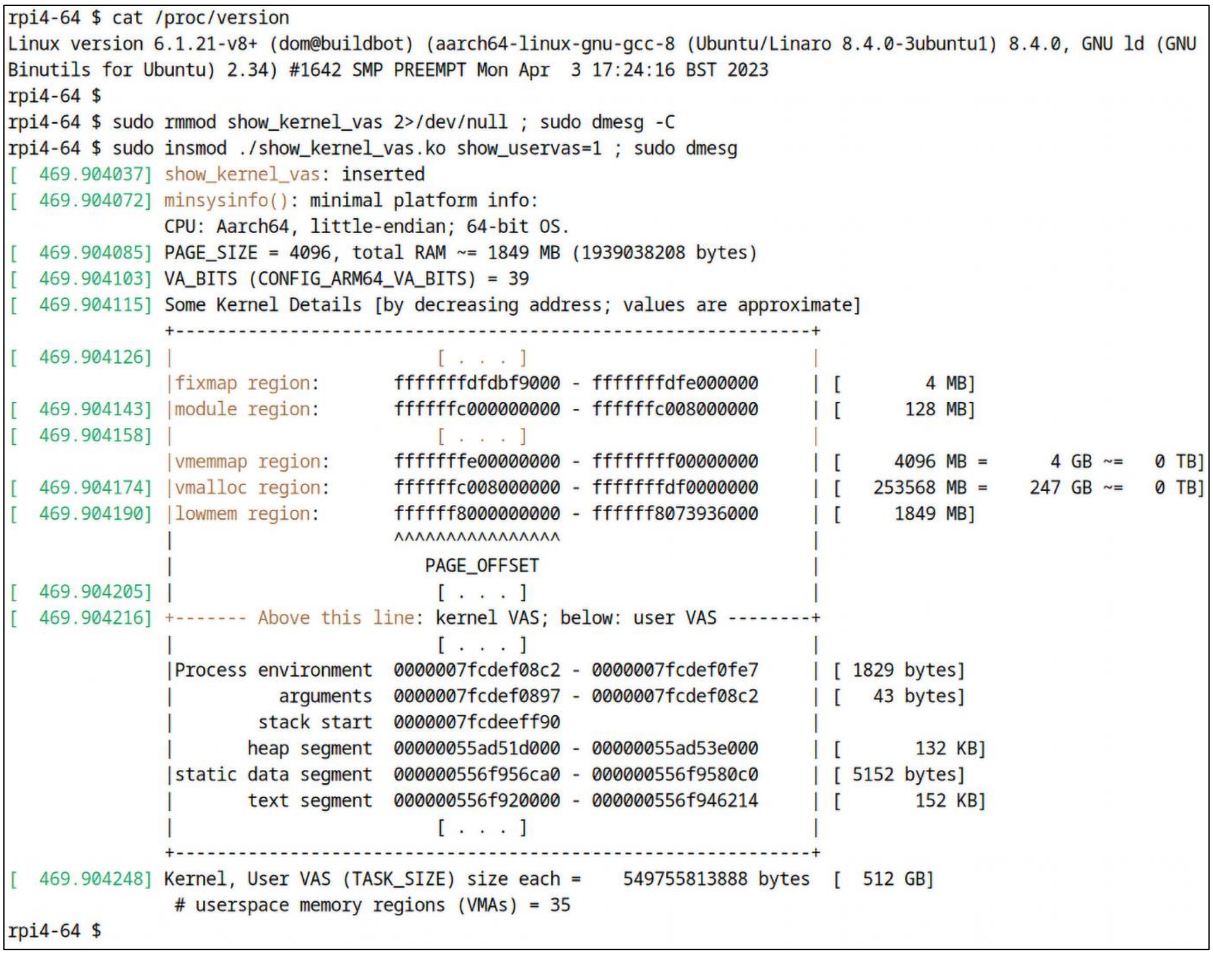
```
rpi0w $ ./procmap --pid=1 --verbose | tee aarch32_rpi0w.txt
[i] will display memory map for process PID=1
[i] running in VERBOSE mode
[v] kernel: init kernel LKM and get details:
[v]  debugfs location verfied
[v]  LKM inserted into kernel
[v]  debugfs file present
[v] Parsing in various kernel variables as required

[v] set config for Aarch32:
Detected machine type: ARM-32, 32-bit OS
--------------------------------------------------
[v] System details detected ::
--------------------------------------------------
VECTORS_BASE = ffff0000
FIXADDR_START = ffc80000
MODULES_VADDR = bf000000
MODULES_END = c0000000
VMALLOC_START = dc800000
VMALLOC_END = ff800000
PAGE_OFFSET = c0000000
TASK_SIZE = bf000000
ARCH = Aarch32
IS_64_BIT = 0
PAGE_SIZE = 4096
KERNEL_VAS_SIZE = 1090519040
USER_VAS_SIZE = 3204448256
HIGHEST_KVA = 0xffffffff
START_KVA = bf000000
START_KVA_DEC = 3204448256
END_UVA = beffffff
END_UVA_DEC = 3204448255
START_UVA = 0x0
--------------------------------------------------
```



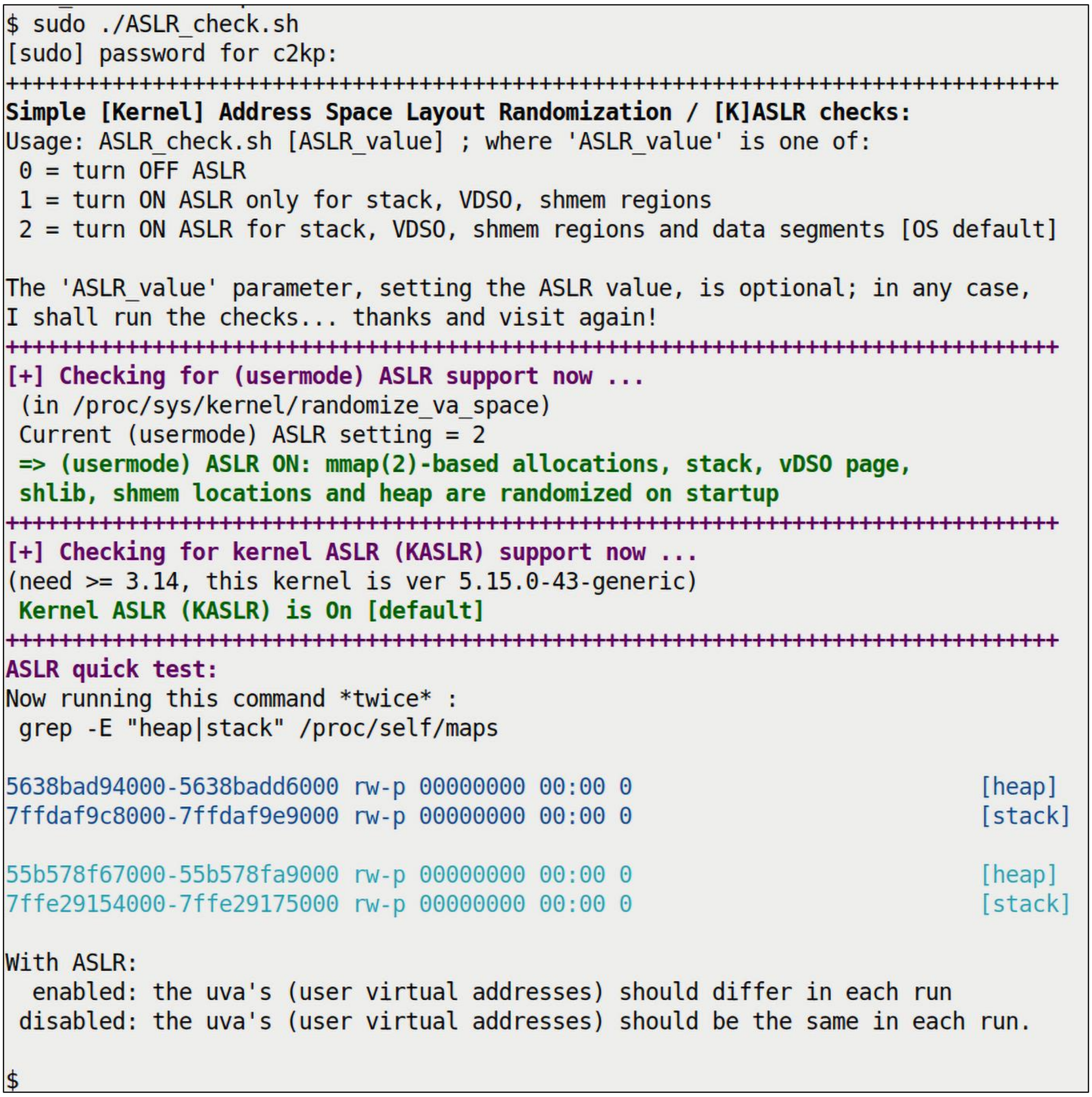
```
VAS mappings:  name    [ size,perms,u:maptype,u:0xfile-offset]
+------------------  K E R N E L   V A S    end kva  ------------------+ ffffffff
|<... K sparse region ...>  [  59 KB,--- ]                             |
+----------------------------------------------------------------------+ ffff1000
|        vector table  [   4 KB,r-- ]                                  |
+----------------------------------------------------------------------+ ffff0000  <-- VECTORS_BASE
|<... K sparse region ...>  [ 960 KB,--- ]                             |
|                                                                      |
+----------------------------------------------------------------------+ fff00000
|       fixmap region [   2.50 MB,r-- ]                                |
|                                                                      |
|                                                                      |
+----------------------------------------------------------------------+ ffc80000  <-- FIXADDR_START
|<... K sparse region ...> [    4.50 MB,--- ]                          |
|                                                                      |
|                                                                      |
+----------------------------------------------------------------------+ ff800000  <-- VMALLOC_END
|      vmalloc region [ 560.00 MB,rw- ]                                |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
+----------------------------------------------------------------------+ dc800000  <-- VMALLOC_START
|<... K sparse region ...> [  26.71 MB,--- ]                           |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
+----------------------------------------------------------------------+ dad4a000
|       lowmem region [ 429.28 MB,rwx ]                                |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
|    [-------------------------------------------------------------]   | c0e29fff
|         Kernel data [   1.99 MB,... ]                                |
|                                                                      |
|                                                                      |
|    [-------------------------------------------------------------]   | c0bbffff
|         Kernel code [  11.71 MB,... ]                                |
|                                                                      |
|                                                                      |
|                                                                      |
|                                                                      |
+----------------------------------------------------------------------+ c0000000  <-- MODULES_END / PAGE_OFFSET
|      module region: [  16.00 MB,rwx ]                                |
|                                                                      |
|                                                                      |
|                                                                      |
+------------------  K E R N E L   V A S  start kva  ------------------+ bf000000
+------------------      U S E R   V A S    end uva  ------------------+ beffffff
|<... Sparse Region ...> [   1.58 MB,---,-,0x0]                        |
|                                                                      |
```



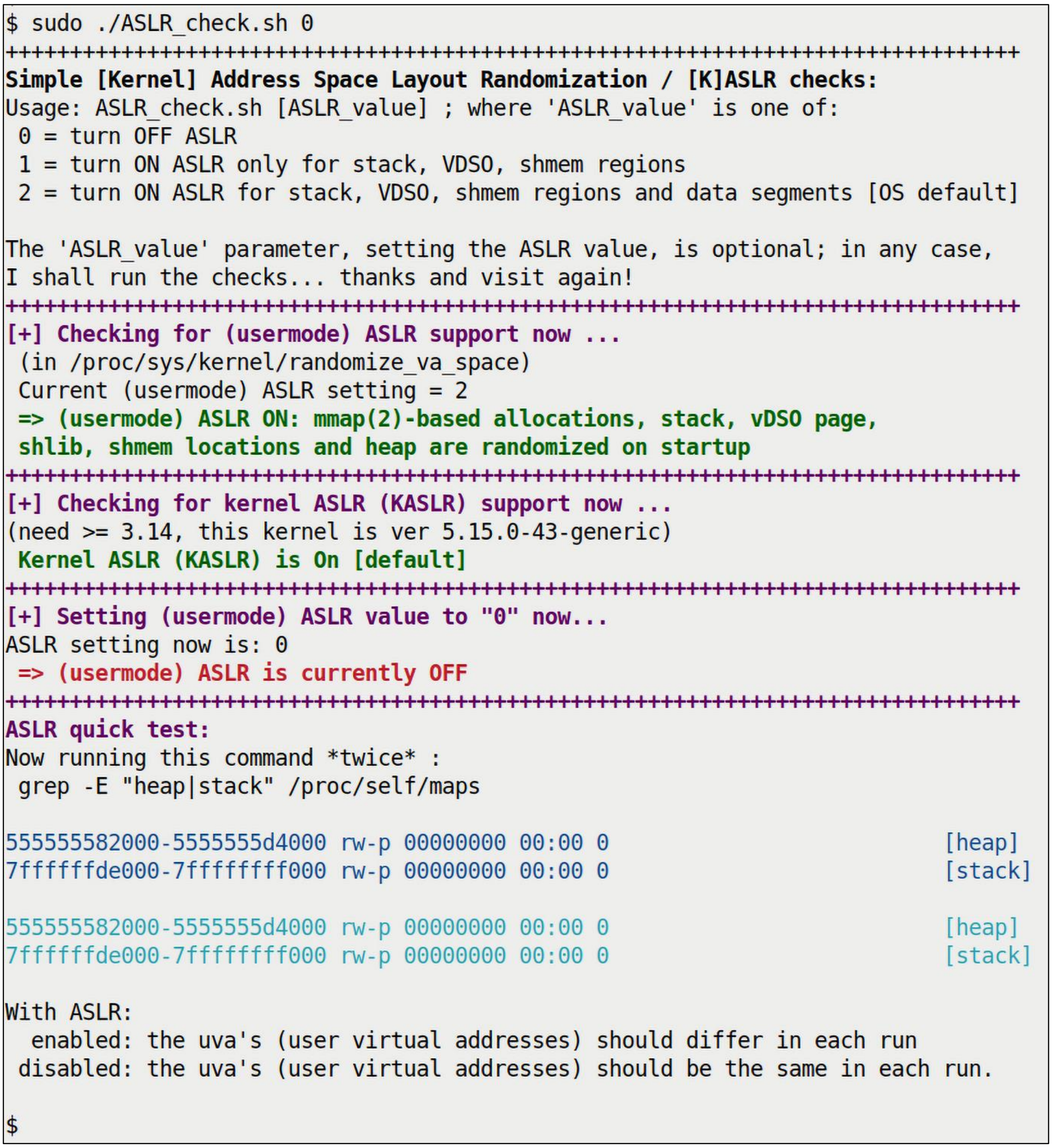
```
rpi0w $ uname -r ; sudo rmmod show_kernel_vas 2>/dev/null ; sudo dmesg -C
6.1.21+
rpi0w $ sudo insmod ./show_kernel_vas.ko show_uservas=1 ; dmesg
[ 7725.559741] show_kernel_vas: inserted
[ 7725.559783] minsysinfo(): minimal platform info:
               CPU: ARM-32, little-endian; 32-bit OS.
[ 7725.559794] PAGE_SIZE = 4096, total RAM ~= 429 MB (450142208 bytes)
[ 7725.559813] Some Kernel Details [by decreasing address; values are approximate]
               +-------------------------------------------------------------+
[ 7725.559822] |                         [ . . . ]                           |
               |vector table:        ffff0000 - ffff1000                     | [    4 KB]
[ 7725.559837] |                         [ . . . ]                           |
               |fixmap region:       ffc80000 - fff00000                     | [    2 MB]
[ 7725.559850] |vmalloc region:      dc800000 - ff800000                     | [  560 MB]
[ 7725.559861] |lowmem region:       c0000000 - dad4a000                     | [  429 MB]
               |                     ^^^^^^^^                                |
               |                    PAGE_OFFSET                              |
[ 7725.559877] |module region:       bf000000 - c0000000                     | [   16 MB]
[ 7725.559888] |                         [ . . . ]                           |
[ 7725.559895] +------- Above this line: kernel VAS; below: user VAS --------+
               |                         [ . . . ]                           |
               |Process environment  bec7f8c8 - bec7ffeb                     | [ 1827 bytes]
               |          arguments  bec7f89d - bec7f8c8                     | [   43 bytes]
               |        stack start  bec7f790                                |
               |       heap segment  01947000 - 01968000                     | [       132 KB]
               |static data segment  00040c44 - 00041038                     | [ 1012 bytes]
               |       text segment  00010000 - 000303d8                     | [       128 KB]
               |                         [ . . . ]                           |
               +-------------------------------------------------------------+
[ 7725.559935] Size of User VAS size (TASK_SIZE) = 3204448256 bytes           [  3056 GB]
                # userspace memory regions (VMAs) = 38
rpi0w $
```



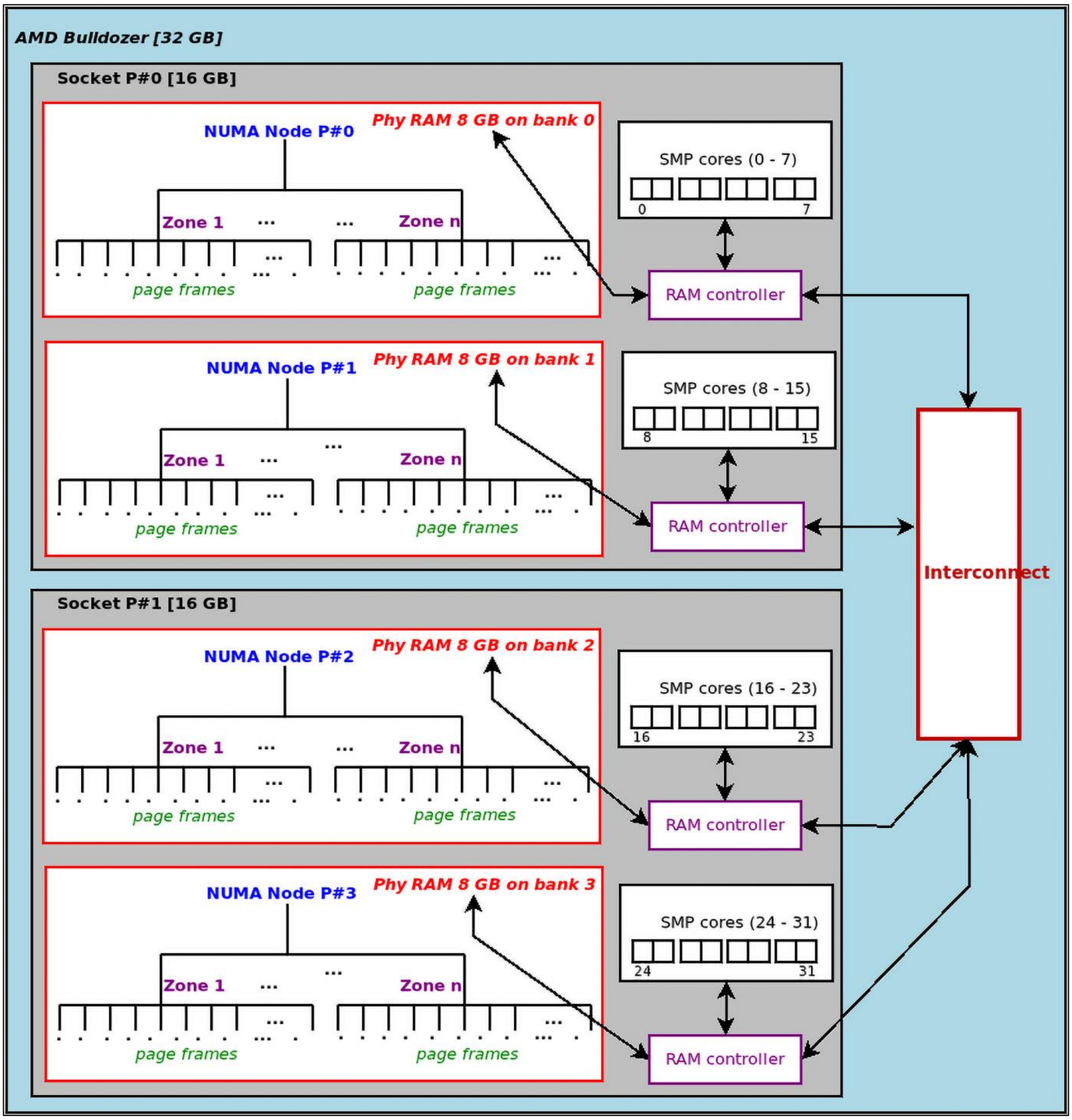
```
rpi4-64 $ cat /proc/version
Linux version 6.1.21-v8+ (dom@buildbot) (aarch64-linux-gnu-gcc-8 (Ubuntu/Linaro 8.4.0-3ubuntu1) 8.4.0, GNU ld (GNU
Binutils for Ubuntu) 2.34) #1642 SMP PREEMPT Mon Apr  3 17:24:16 BST 2023
rpi4-64 $
rpi4-64 $ sudo rmmod show_kernel_vas 2>/dev/null ; sudo dmesg -C
rpi4-64 $ sudo insmod ./show_kernel_vas.ko show_uservas=1 ; sudo dmesg
[  469.904037] show_kernel_vas: inserted
[  469.904072] minsysinfo(): minimal platform info:
               CPU: Aarch64, little-endian; 64-bit OS.
[  469.904085] PAGE_SIZE = 4096, total RAM ~= 1849 MB (1939038208 bytes)
[  469.904103] VA_BITS (CONFIG_ARM64_VA_BITS) = 39
[  469.904115] Some Kernel Details [by decreasing address; values are approximate]
               +--------------------------------------------------------------+
[  469.904126] |                         [ . . . ]                            |
               |fixmap region:       fffffffdfdbf9000 - fffffffdfe000000      | [         4 MB]
[  469.904143] |module region:       ffffffc000000000 - ffffffc008000000      | [       128 MB]
[  469.904158] |                         [ . . . ]                            |
               |vmemmap region:      fffffffe00000000 - ffffffff00000000      | [     4096 MB =       4 GB ~=   0 TB]
[  469.904174] |vmalloc region:      ffffffc008000000 - fffffffdf0000000      | [   253568 MB =     247 GB ~=   0 TB]
[  469.904190] |lowmem region:       ffffff8000000000 - ffffff8073936000      | [     1849 MB]
               |                     ^^^^^^^^^^^^^^^^                         |
               |                        PAGE_OFFSET                           |
[  469.904205] |                         [ . . . ]                            |
[  469.904216] +------- Above this line: kernel VAS; below: user VAS --------+
               |                         [ . . . ]                            |
               |Process environment  0000007fcdef08c2 - 0000007fcdef0fe7      | [ 1829 bytes]
               |          arguments  0000007fcdef0897 - 0000007fcdef08c2      | [   43 bytes]
               |        stack start  0000007fcdeeff90                         |
               |       heap segment  00000055ad51d000 - 00000055ad53e000      | [        132 KB]
               |static data segment  000000556f956ca0 - 000000556f9580c0      | [ 5152 bytes]
               |       text segment  000000556f920000 - 000000556f946214      | [        152 KB]
               |                         [ . . . ]                            |
               +--------------------------------------------------------------+
[  469.904248] Kernel, User VAS (TASK_SIZE) size each =    549755813888 bytes  [  512 GB]
                # userspace memory regions (VMAs) = 35
rpi4-64 $
```

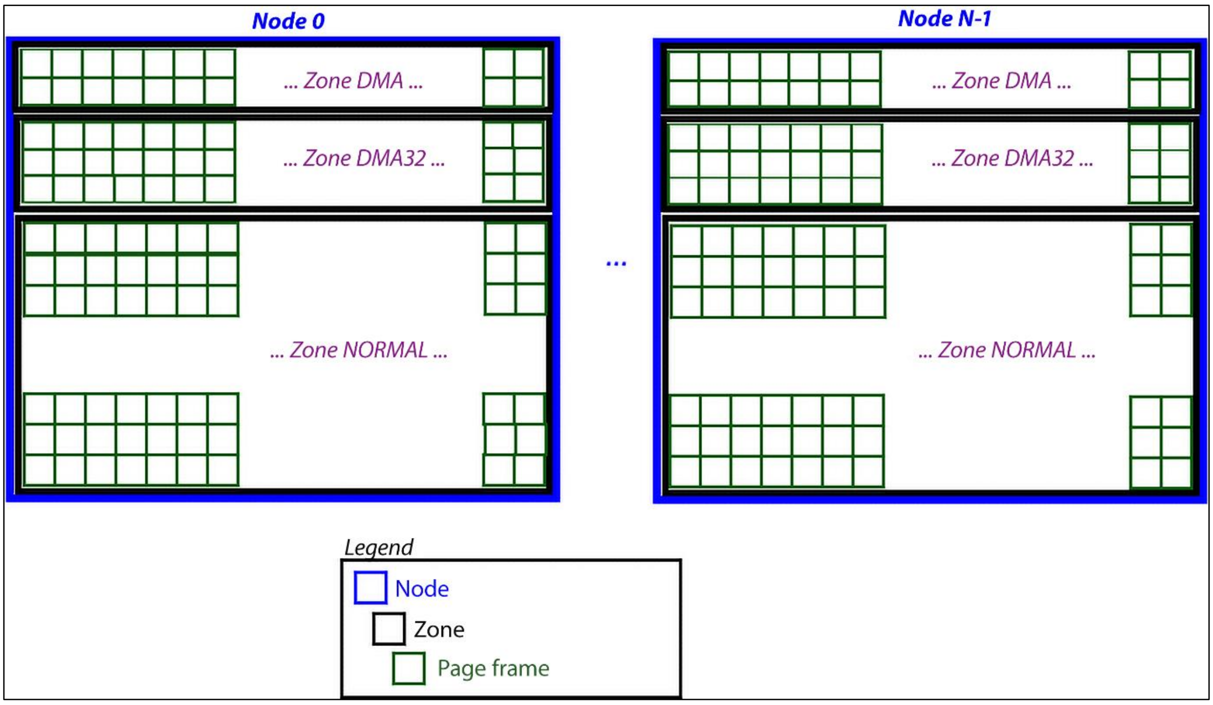


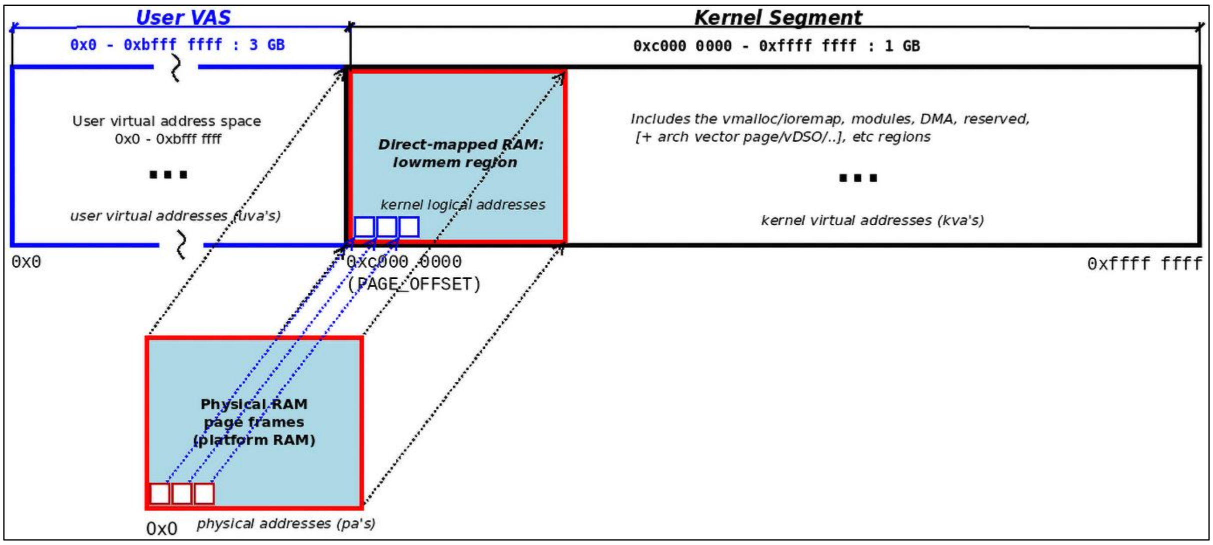
```
$ sudo ./ASLR_check.sh
[sudo] password for c2kp:
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Simple [Kernel] Address Space Layout Randomization / [K]ASLR checks:
Usage: ASLR_check.sh [ASLR_value] ; where 'ASLR_value' is one of:
 0 = turn OFF ASLR
 1 = turn ON ASLR only for stack, VDSO, shmem regions
 2 = turn ON ASLR for stack, VDSO, shmem regions and data segments [OS default]

The 'ASLR_value' parameter, setting the ASLR value, is optional; in any case,
I shall run the checks... thanks and visit again!
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
[+] Checking for (usermode) ASLR support now ...
 (in /proc/sys/kernel/randomize_va_space)
 Current (usermode) ASLR setting = 2
 => (usermode) ASLR ON: mmap(2)-based allocations, stack, vDSO page,
 shlib, shmem locations and heap are randomized on startup
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
[+] Checking for kernel ASLR (KASLR) support now ...
(need >= 3.14, this kernel is ver 5.15.0-43-generic)
 Kernel ASLR (KASLR) is On [default]
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
ASLR quick test:
Now running this command *twice* :
 grep -E "heap|stack" /proc/self/maps

5638bad94000-5638badd6000 rw-p 00000000 00:00 0                          [heap]
7ffdaf9c8000-7ffdaf9e9000 rw-p 00000000 00:00 0                          [stack]

55b578f67000-55b578fa9000 rw-p 00000000 00:00 0                          [heap]
7ffe29154000-7ffe29175000 rw-p 00000000 00:00 0                          [stack]

With ASLR:
  enabled: the uva's (user virtual addresses) should differ in each run
 disabled: the uva's (user virtual addresses) should be the same in each run.

$
```



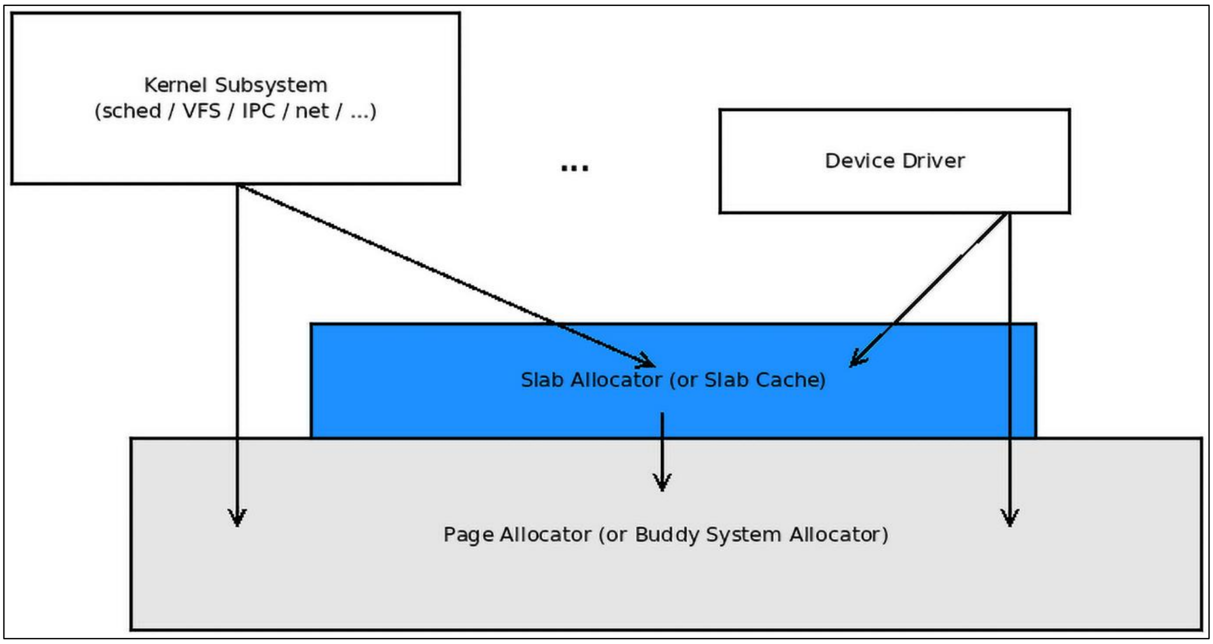
```
$ sudo ./ASLR_check.sh 0
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Simple [Kernel] Address Space Layout Randomization / [K]ASLR checks:
Usage: ASLR_check.sh [ASLR_value] ; where 'ASLR_value' is one of:
 0 = turn OFF ASLR
 1 = turn ON ASLR only for stack, VDSO, shmem regions
 2 = turn ON ASLR for stack, VDSO, shmem regions and data segments [OS default]

The 'ASLR_value' parameter, setting the ASLR value, is optional; in any case,
I shall run the checks... thanks and visit again!
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
[+] Checking for (usermode) ASLR support now ...
 (in /proc/sys/kernel/randomize_va_space)
 Current (usermode) ASLR setting = 2
 => (usermode) ASLR ON: mmap(2)-based allocations, stack, vDSO page,
 shlib, shmem locations and heap are randomized on startup
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
[+] Checking for kernel ASLR (KASLR) support now ...
(need >= 3.14, this kernel is ver 5.15.0-43-generic)
 Kernel ASLR (KASLR) is On [default]
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
[+] Setting (usermode) ASLR value to "0" now...
ASLR setting now is: 0
 => (usermode) ASLR is currently OFF
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
ASLR quick test:
Now running this command *twice* :
 grep -E "heap|stack" /proc/self/maps

555555582000-5555555d4000 rw-p 00000000 00:00 0                          [heap]
7ffffffde000-7ffffffff000 rw-p 00000000 00:00 0                          [stack]

555555582000-5555555d4000 rw-p 00000000 00:00 0                          [heap]
7ffffffde000-7ffffffff000 rw-p 00000000 00:00 0                          [stack]

With ASLR:
  enabled: the uva's (user virtual addresses) should differ in each run
 disabled: the uva's (user virtual addresses) should be the same in each run.

$
```

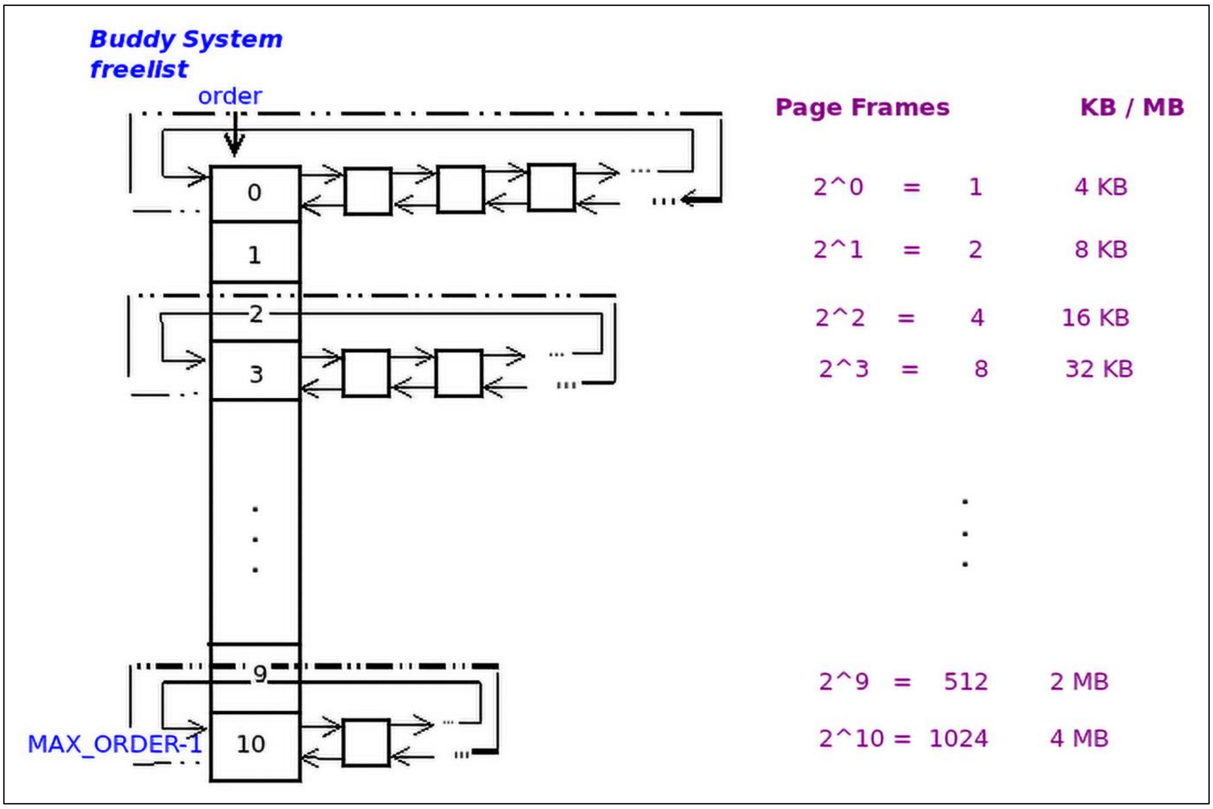

AMD Bulldozer [32 GB]
Socket P#0 [16 GB]
NUMA Node P#0
Phy RAM 8 GB on bank 0
Zone 1
Zone n
page frames
page frames
SMP cores (0 - 7)
0
7
RAM controller
NUMA Node P#1
Phy RAM 8 GB on bank 1
Zone 1
Zone n
page frames
page frames
SMP cores (8 - 15)
8
15
RAM controller
Interconnect
Socket P#1 [16 GB]
NUMA Node P#2
Phy RAM 8 GB on bank 2
Zone 1
Zone n
page frames
page frames
SMP cores (16 - 23)
16
23
RAM controller
NUMA Node P#3
Phy RAM 8 GB on bank 3
Zone 1
Zone n
page frames
page frames
SMP cores (24 - 31)
24
31
RAM controller

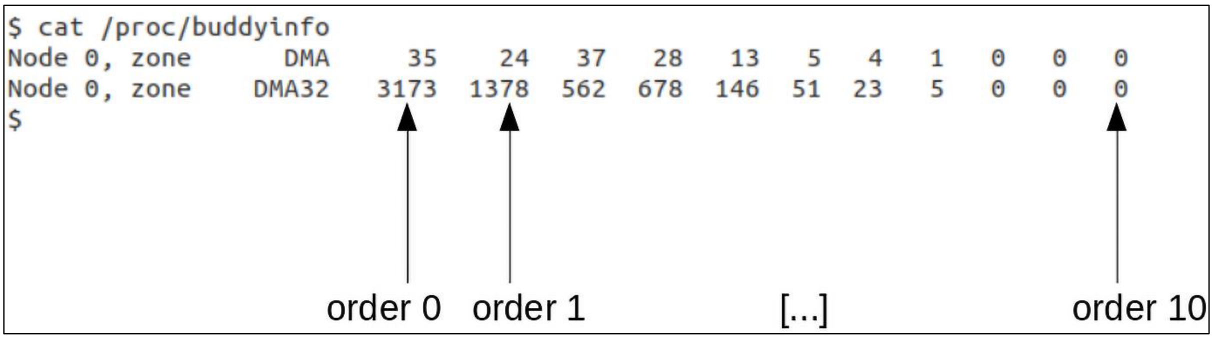

Node 0
... Zone DMA ...
... Zone DMA32 ...
... Zone NORMAL ...
...
Node N-1
... Zone DMA ...
... Zone DMA32 ...
... Zone NORMAL ...
Legend
Node
Zone
Page frame


User VAS
0x0 - 0xbfff ffff : 3 GB
Kernel Segment
0xc000 0000 - 0xffff ffff : 1 GB
User virtual address space
0x0 - 0xbfff ffff
...
user virtual addresses (uva's)
Direct-mapped RAM: lowmem region
kernel logical addresses
Includes the vmalloc/ioremap, modules, DMA, reserved, [+ arch vector page/vDSO/..], etc regions
...
kernel virtual addresses (kva's)
0x0
0xc000 0000
(PAGE_OFFSET)
0xffff ffff
Physical RAM page frames (platform RAM)
0x0
physical addresses (pa's)

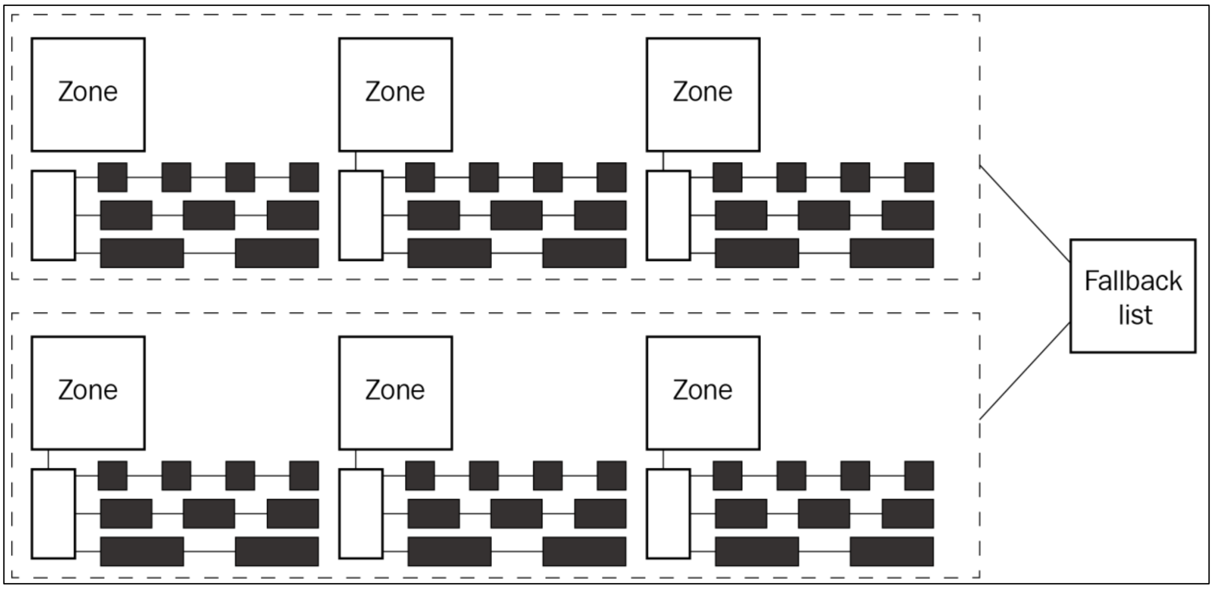
Chapter 8: Kernel Memory Allocation for Module Authors – Part 1

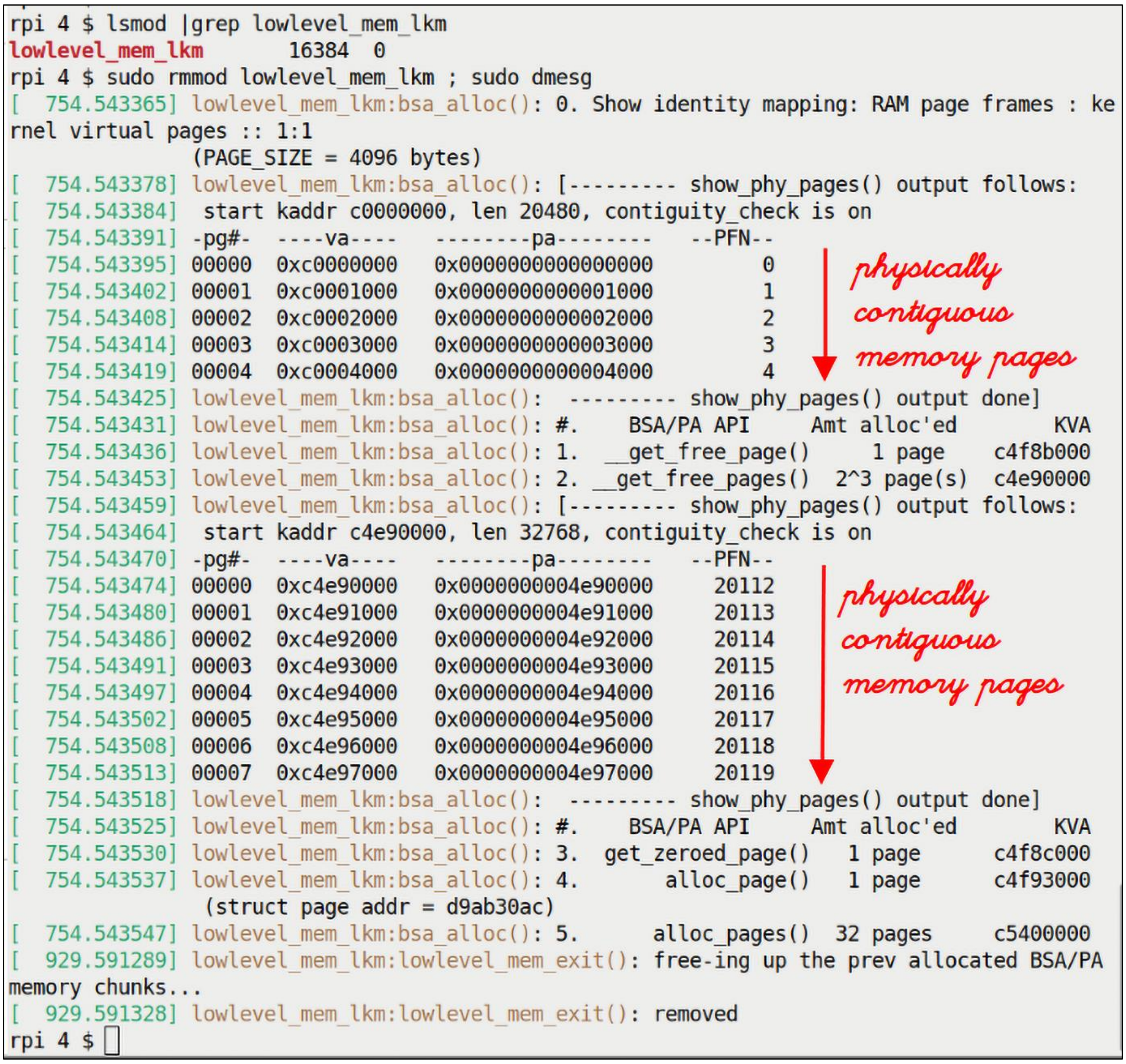

Kernel Subsystem
(sched / VFS / IPC / net / ...)
...
Device Driver
Slab Allocator (or Slab Cache)
Page Allocator (or Buddy System Allocator)





```
$ cat /proc/buddyinfo
Node 0, zone      DMA     35     24    37    28    13     5     4     1     0     0     0
Node 0, zone    DMA32   3173   1378   562   678   146    51    23     5     0     0     0
$
```

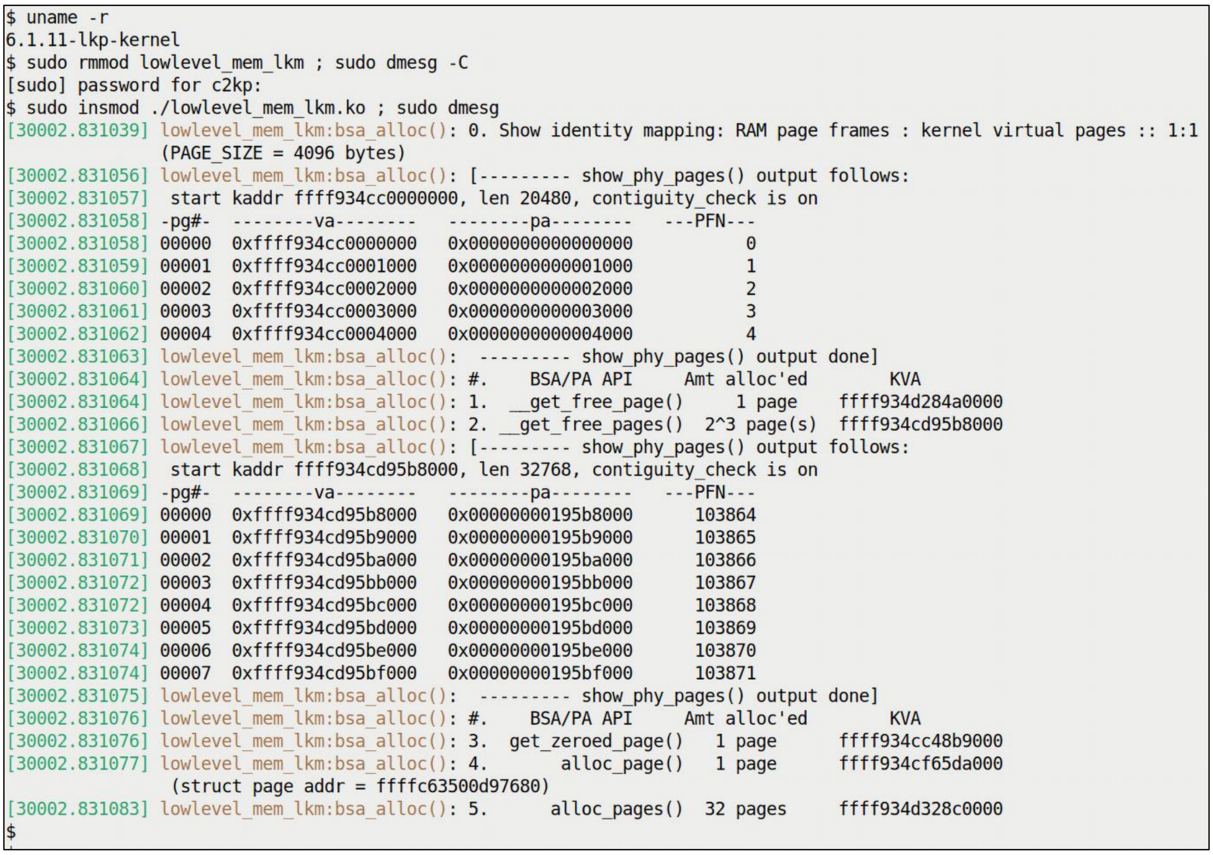




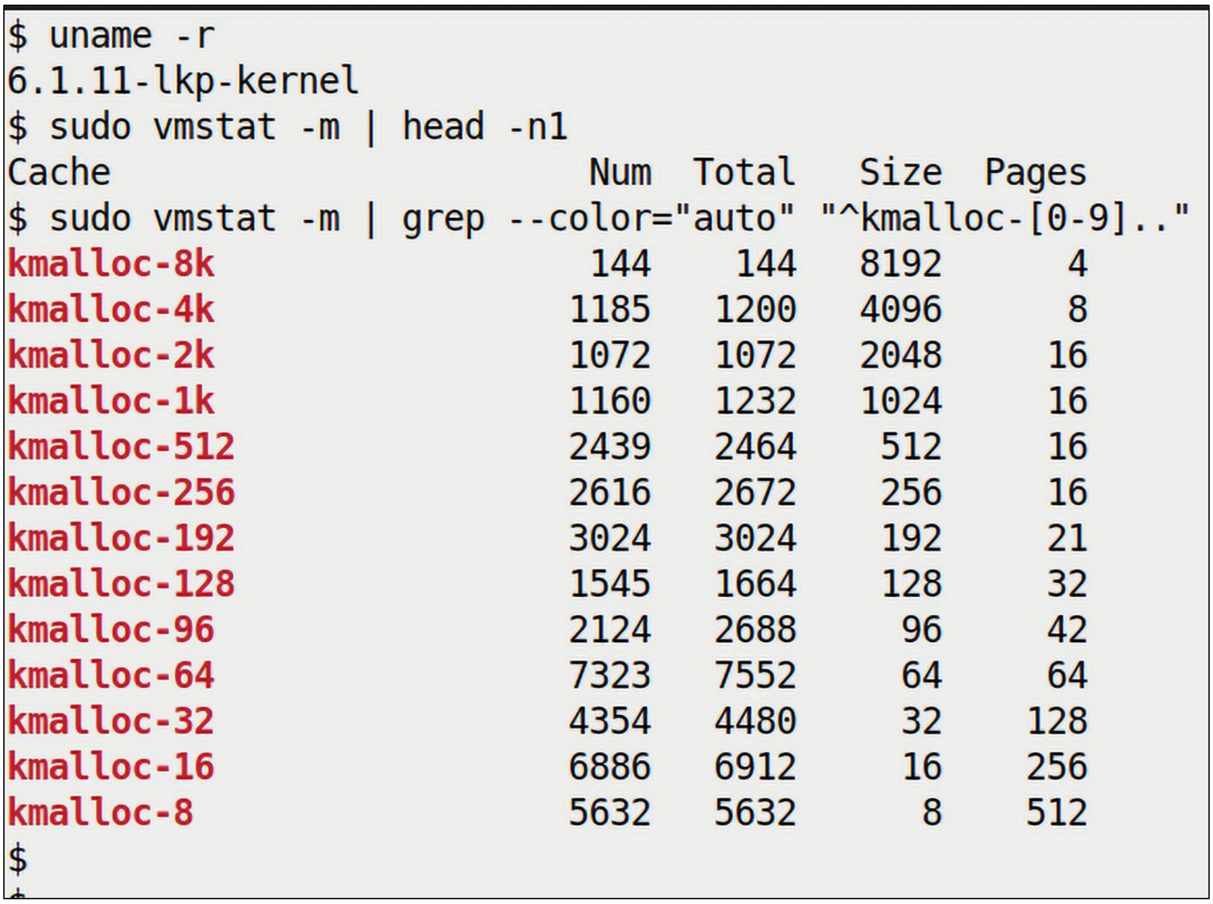
```
rpi 4 $ lsmod |grep lowlevel_mem_lkm
lowlevel_mem_lkm       16384  0
rpi 4 $ sudo rmmod lowlevel_mem_lkm ; sudo dmesg
[  754.543365] lowlevel_mem_lkm:bsa_alloc(): 0. Show identity mapping: RAM page frames : ke
rnel virtual pages :: 1:1
               (PAGE_SIZE = 4096 bytes)
[  754.543378] lowlevel_mem_lkm:bsa_alloc(): [--------- show_phy_pages() output follows:
[  754.543384]  start kaddr c0000000, len 20480, contiguity_check is on
[  754.543391] -pg#-   ----va----    --------pa--------    --PFN--
[  754.543395] 00000  0xc0000000   0x0000000000000000         0
[  754.543402] 00001  0xc0001000   0x0000000000001000         1
[  754.543408] 00002  0xc0002000   0x0000000000002000         2
[  754.543414] 00003  0xc0003000   0x0000000000003000         3
[  754.543419] 00004  0xc0004000   0x0000000000004000         4
[  754.543425] lowlevel_mem_lkm:bsa_alloc():  --------- show_phy_pages() output done]
[  754.543431] lowlevel_mem_lkm:bsa_alloc(): #.    BSA/PA API      Amt alloc'ed        KVA
[  754.543436] lowlevel_mem_lkm:bsa_alloc(): 1.  __get_free_page()      1 page     c4f8b000
[  754.543453] lowlevel_mem_lkm:bsa_alloc(): 2. __get_free_pages()  2^3 page(s)  c4e90000
[  754.543459] lowlevel_mem_lkm:bsa_alloc(): [--------- show_phy_pages() output follows:
[  754.543464]  start kaddr c4e90000, len 32768, contiguity_check is on
[  754.543470] -pg#-   ----va----    --------pa--------    --PFN--
[  754.543474] 00000  0xc4e90000   0x0000000004e90000     20112
[  754.543480] 00001  0xc4e91000   0x0000000004e91000     20113
[  754.543486] 00002  0xc4e92000   0x0000000004e92000     20114
[  754.543491] 00003  0xc4e93000   0x0000000004e93000     20115
[  754.543497] 00004  0xc4e94000   0x0000000004e94000     20116
[  754.543502] 00005  0xc4e95000   0x0000000004e95000     20117
[  754.543508] 00006  0xc4e96000   0x0000000004e96000     20118
[  754.543513] 00007  0xc4e97000   0x0000000004e97000     20119
[  754.543518] lowlevel_mem_lkm:bsa_alloc():  --------- show_phy_pages() output done]
[  754.543525] lowlevel_mem_lkm:bsa_alloc(): #.    BSA/PA API      Amt alloc'ed        KVA
[  754.543530] lowlevel_mem_lkm:bsa_alloc(): 3.  get_zeroed_page()   1 page      c4f8c000
[  754.543537] lowlevel_mem_lkm:bsa_alloc(): 4.      alloc_page()   1 page      c4f93000
                (struct page addr = d9ab30ac)
[  754.543547] lowlevel_mem_lkm:bsa_alloc(): 5.     alloc_pages()  32 pages      c5400000
[  929.591289] lowlevel_mem_lkm:lowlevel_mem_exit(): free-ing up the prev allocated BSA/PA
memory chunks...
[  929.591328] lowlevel_mem_lkm:lowlevel_mem_exit(): removed
rpi 4 $ 
```

physically
contiguous
memory pages

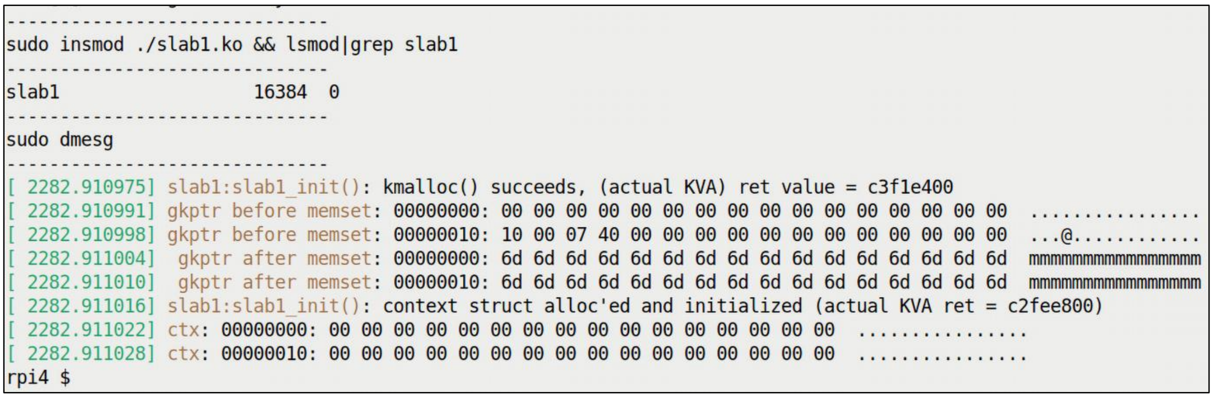
physically
contiguous
memory pages



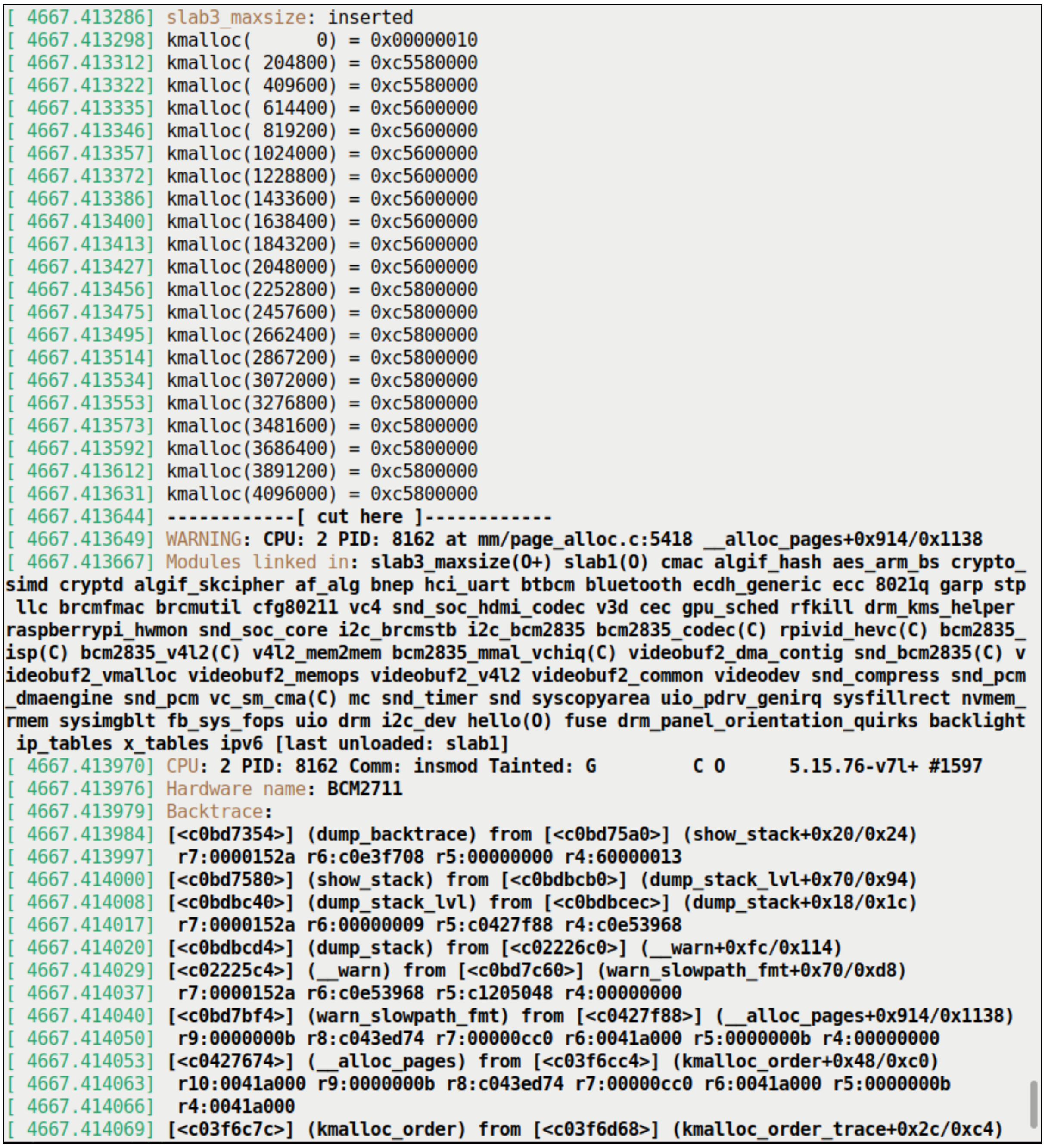
```
$ uname -r
6.1.11-lkp-kernel
$ sudo rmmod lowlevel_mem_lkm ; sudo dmesg -C
[sudo] password for c2kp:
$ sudo insmod ./lowlevel_mem_lkm.ko ; sudo dmesg
[30002.831039] lowlevel_mem_lkm:bsa_alloc(): 0. Show identity mapping: RAM page frames : kernel virtual pages :: 1:1
               (PAGE_SIZE = 4096 bytes)
[30002.831056] lowlevel_mem_lkm:bsa_alloc(): [--------- show_phy_pages() output follows:
[30002.831057]  start kaddr ffff934cc0000000, len 20480, contiguity_check is on
[30002.831058] -pg#-  --------va--------   --------pa--------   ---PFN---
[30002.831058] 00000  0xffff934cc0000000   0x0000000000000000          0
[30002.831059] 00001  0xffff934cc0001000   0x0000000000001000          1
[30002.831060] 00002  0xffff934cc0002000   0x0000000000002000          2
[30002.831061] 00003  0xffff934cc0003000   0x0000000000003000          3
[30002.831062] 00004  0xffff934cc0004000   0x0000000000004000          4
[30002.831063] lowlevel_mem_lkm:bsa_alloc():  --------- show_phy_pages() output done]
[30002.831064] lowlevel_mem_lkm:bsa_alloc(): #.    BSA/PA API     Amt alloc'ed        KVA
[30002.831064] lowlevel_mem_lkm:bsa_alloc(): 1.  __get_free_page()     1 page    ffff934d284a0000
[30002.831066] lowlevel_mem_lkm:bsa_alloc(): 2. __get_free_pages()  2^3 page(s)  ffff934cd95b8000
[30002.831067] lowlevel_mem_lkm:bsa_alloc(): [--------- show_phy_pages() output follows:
[30002.831068]  start kaddr ffff934cd95b8000, len 32768, contiguity_check is on
[30002.831069] -pg#-  --------va--------   --------pa--------   ---PFN---
[30002.831069] 00000  0xffff934cd95b8000   0x00000000195b8000      103864
[30002.831070] 00001  0xffff934cd95b9000   0x00000000195b9000      103865
[30002.831071] 00002  0xffff934cd95ba000   0x00000000195ba000      103866
[30002.831072] 00003  0xffff934cd95bb000   0x00000000195bb000      103867
[30002.831072] 00004  0xffff934cd95bc000   0x00000000195bc000      103868
[30002.831073] 00005  0xffff934cd95bd000   0x00000000195bd000      103869
[30002.831074] 00006  0xffff934cd95be000   0x00000000195be000      103870
[30002.831074] 00007  0xffff934cd95bf000   0x00000000195bf000      103871
[30002.831075] lowlevel_mem_lkm:bsa_alloc():  --------- show_phy_pages() output done]
[30002.831076] lowlevel_mem_lkm:bsa_alloc(): #.    BSA/PA API     Amt alloc'ed        KVA
[30002.831076] lowlevel_mem_lkm:bsa_alloc(): 3.  get_zeroed_page()   1 page      ffff934cc48b9000
[30002.831077] lowlevel_mem_lkm:bsa_alloc(): 4.       alloc_page()   1 page      ffff934cf65da000
                (struct page addr = ffffc63500d97680)
[30002.831083] lowlevel_mem_lkm:bsa_alloc(): 5.      alloc_pages()  32 pages     ffff934d328c0000
$
```



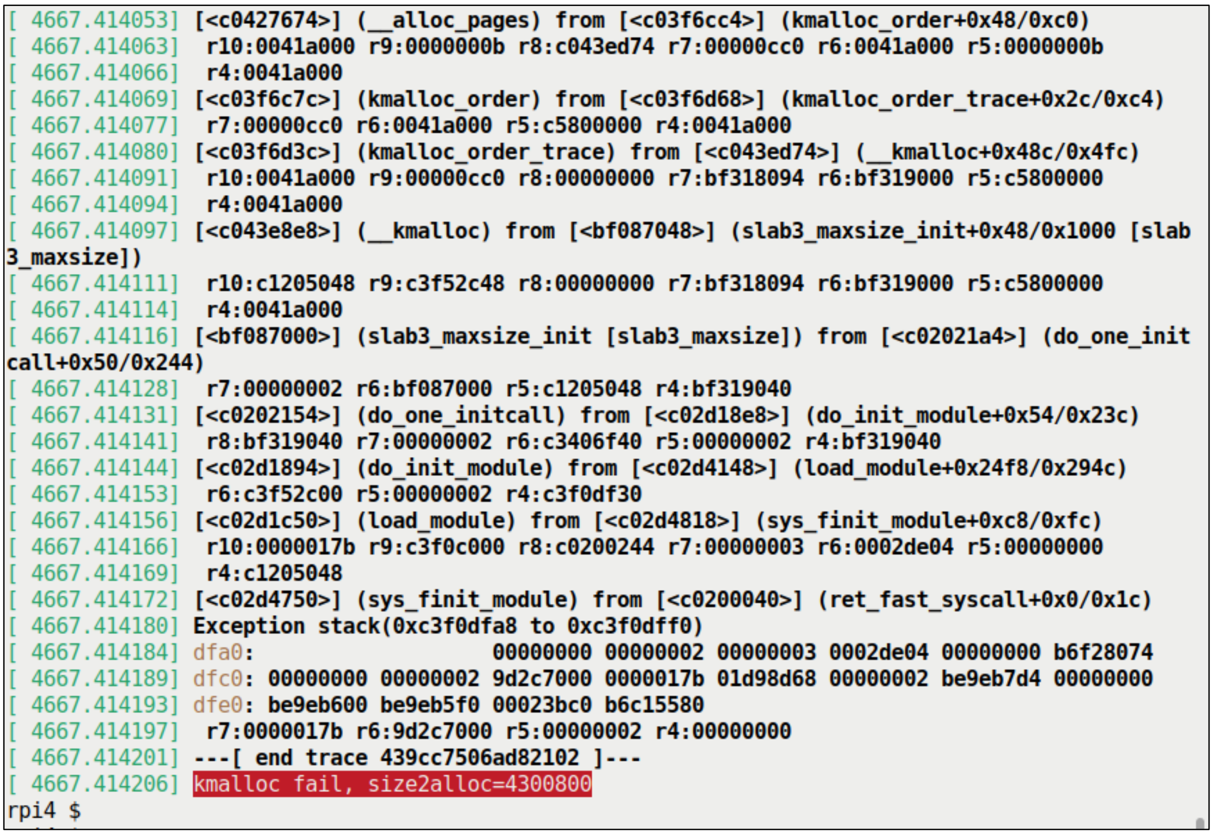
```
$ uname -r
6.1.11-lkp-kernel
$ sudo vmstat -m | head -n1
Cache                       Num  Total   Size  Pages
$ sudo vmstat -m | grep --color="auto" "^kmalloc-[0-9].."
kmalloc-8k                  144    144   8192      4
kmalloc-4k                 1185   1200   4096      8
kmalloc-2k                 1072   1072   2048     16
kmalloc-1k                 1160   1232   1024     16
kmalloc-512                2439   2464    512     16
kmalloc-256                2616   2672    256     16
kmalloc-192                3024   3024    192     21
kmalloc-128                1545   1664    128     32
kmalloc-96                 2124   2688     96     42
kmalloc-64                 7323   7552     64     64
kmalloc-32                 4354   4480     32    128
kmalloc-16                 6886   6912     16    256
kmalloc-8                  5632   5632      8    512
$
```



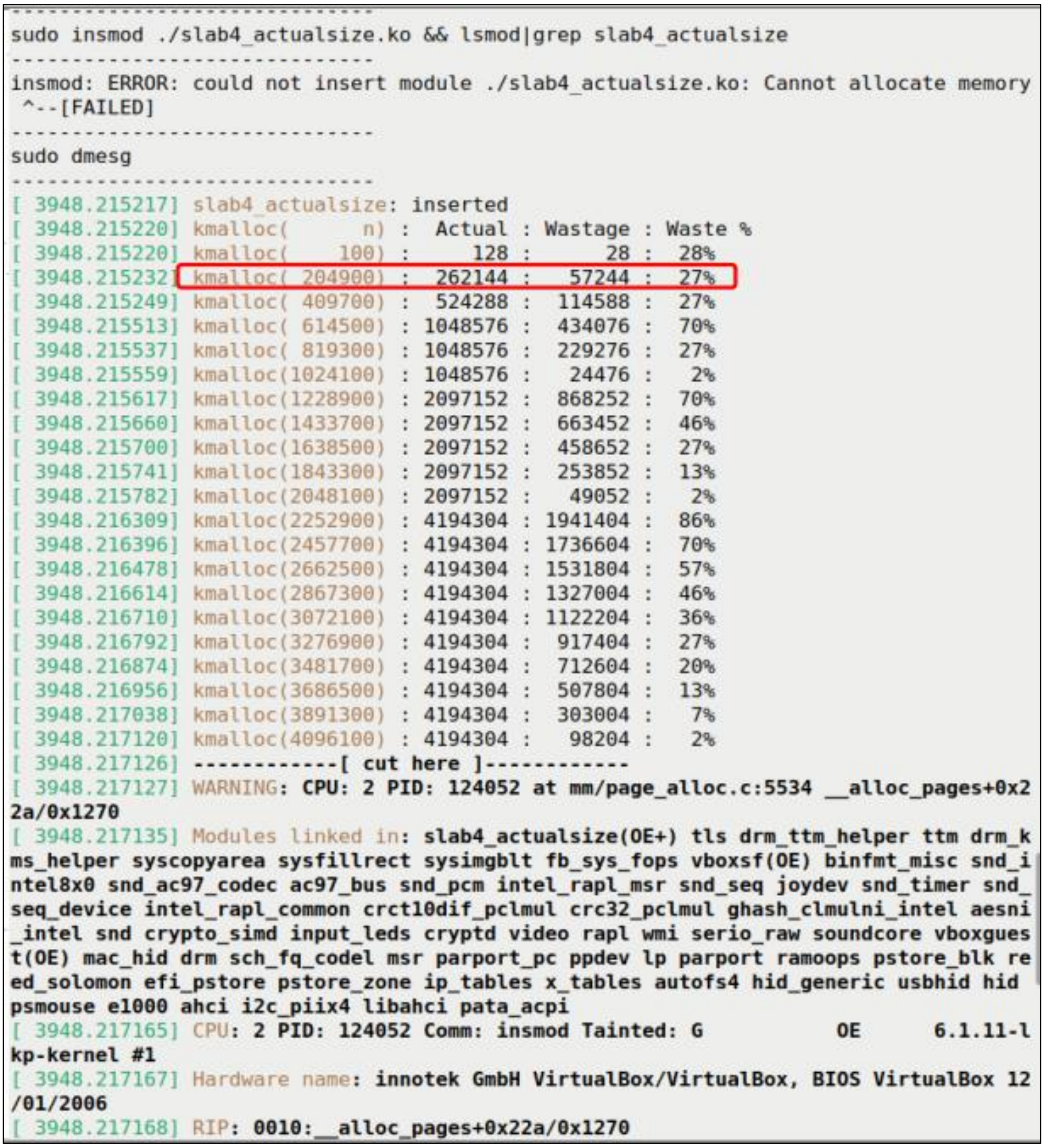
```
------------------------------
sudo insmod ./slab1.ko && lsmod|grep slab1
------------------------------
slab1                  16384  0
------------------------------
sudo dmesg
------------------------------
[ 2282.910975] slab1:slab1_init(): kmalloc() succeeds, (actual KVA) ret value = c3f1e400
[ 2282.910991] gkptr before memset: 00000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[ 2282.910998] gkptr before memset: 00000010: 10 00 07 40 00 00 00 00 00 00 00 00 00 00 00 00  ...@............
[ 2282.911004]  gkptr after memset: 00000000: 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d  mmmmmmmmmmmmmmmm
[ 2282.911010]  gkptr after memset: 00000010: 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d  mmmmmmmmmmmmmmmm
[ 2282.911016] slab1:slab1_init(): context struct alloc'ed and initialized (actual KVA ret = c2fee800)
[ 2282.911022] ctx: 00000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[ 2282.911028] ctx: 00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
rpi4 $
```



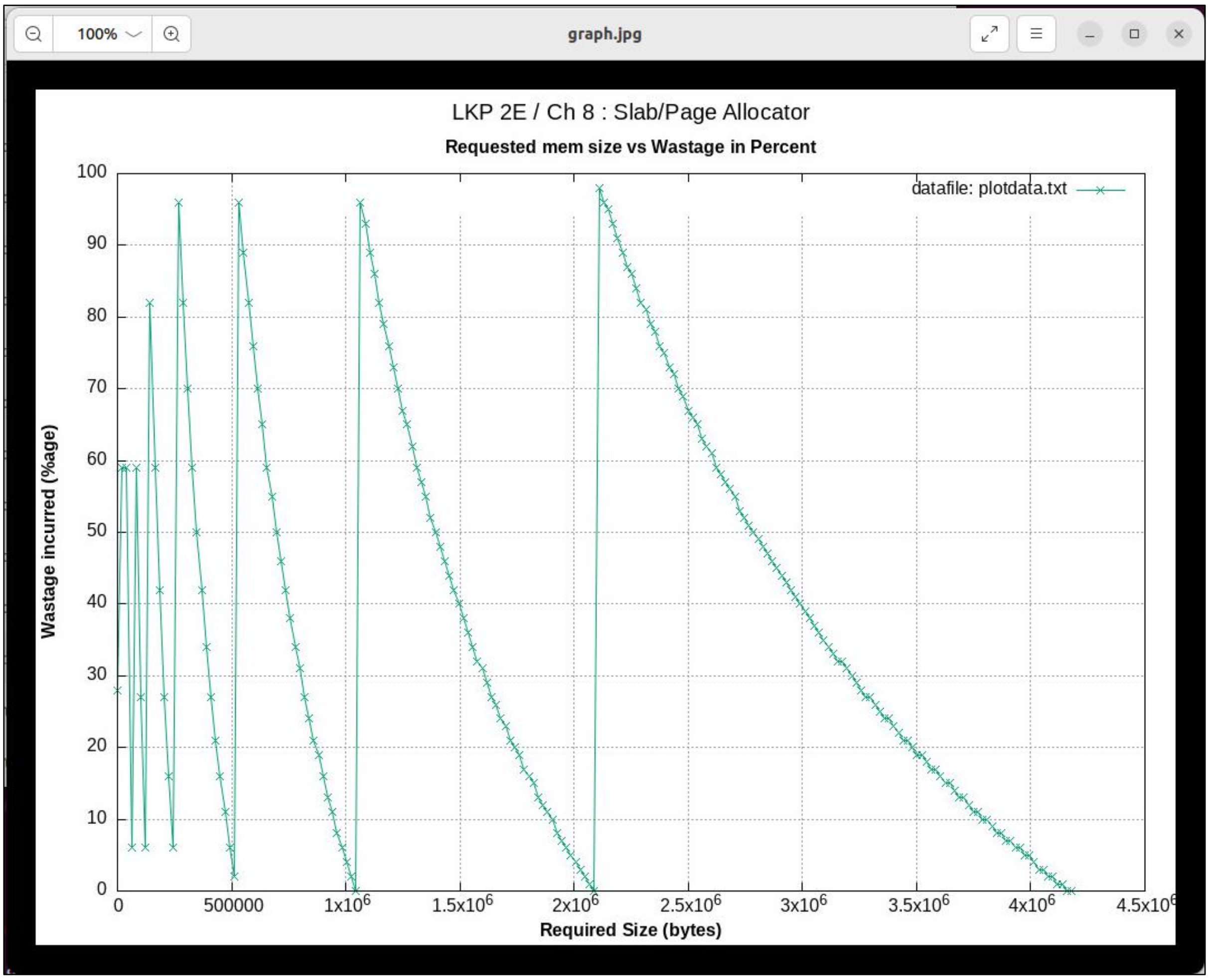
```
[ 4667.413286] slab3_maxsize: inserted
[ 4667.413298] kmalloc(      0) = 0x00000010
[ 4667.413312] kmalloc( 204800) = 0xc5580000
[ 4667.413322] kmalloc( 409600) = 0xc5580000
[ 4667.413335] kmalloc( 614400) = 0xc5600000
[ 4667.413346] kmalloc( 819200) = 0xc5600000
[ 4667.413357] kmalloc(1024000) = 0xc5600000
[ 4667.413372] kmalloc(1228800) = 0xc5600000
[ 4667.413386] kmalloc(1433600) = 0xc5600000
[ 4667.413400] kmalloc(1638400) = 0xc5600000
[ 4667.413413] kmalloc(1843200) = 0xc5600000
[ 4667.413427] kmalloc(2048000) = 0xc5600000
[ 4667.413456] kmalloc(2252800) = 0xc5800000
[ 4667.413475] kmalloc(2457600) = 0xc5800000
[ 4667.413495] kmalloc(2662400) = 0xc5800000
[ 4667.413514] kmalloc(2867200) = 0xc5800000
[ 4667.413534] kmalloc(3072000) = 0xc5800000
[ 4667.413553] kmalloc(3276800) = 0xc5800000
[ 4667.413573] kmalloc(3481600) = 0xc5800000
[ 4667.413592] kmalloc(3686400) = 0xc5800000
[ 4667.413612] kmalloc(3891200) = 0xc5800000
[ 4667.413631] kmalloc(4096000) = 0xc5800000
[ 4667.413644] ------------[ cut here ]------------
[ 4667.413649] WARNING: CPU: 2 PID: 8162 at mm/page_alloc.c:5418 __alloc_pages+0x914/0x1138
[ 4667.413667] Modules linked in: slab3_maxsize(O+) slab1(O) cmac algif_hash aes_arm_bs crypto_
simd cryptd algif_skcipher af_alg bnep hci_uart btbcm bluetooth ecdh_generic ecc 8021q garp stp
 llc brcmfmac brcmutil cfg80211 vc4 snd_soc_hdmi_codec v3d cec gpu_sched rfkill drm_kms_helper
raspberrypi_hwmon snd_soc_core i2c_brcmstb i2c_bcm2835 bcm2835_codec(C) rpivid_hevc(C) bcm2835_
isp(C) bcm2835_v4l2(C) v4l2_mem2mem bcm2835_mmal_vchiq(C) videobuf2_dma_contig snd_bcm2835(C) v
ideobuf2_vmalloc videobuf2_memops videobuf2_v4l2 videobuf2_common videodev snd_compress snd_pcm
_dmaengine snd_pcm vc_sm_cma(C) mc snd_timer snd syscopyarea uio_pdrv_genirq sysfillrect nvmem_
rmem sysimgblt fb_sys_fops uio drm i2c_dev hello(O) fuse drm_panel_orientation_quirks backlight
 ip_tables x_tables ipv6 [last unloaded: slab1]
[ 4667.413970] CPU: 2 PID: 8162 Comm: insmod Tainted: G         C O      5.15.76-v7l+ #1597
[ 4667.413976] Hardware name: BCM2711
[ 4667.413979] Backtrace:
[ 4667.413984] [<c0bd7354>] (dump_backtrace) from [<c0bd75a0>] (show_stack+0x20/0x24)
[ 4667.413997]  r7:0000152a r6:c0e3f708 r5:00000000 r4:60000013
[ 4667.414000] [<c0bd7580>] (show_stack) from [<c0bdbcb0>] (dump_stack_lvl+0x70/0x94)
[ 4667.414008] [<c0bdbc40>] (dump_stack_lvl) from [<c0bdbcec>] (dump_stack+0x18/0x1c)
[ 4667.414017]  r7:0000152a r6:00000009 r5:c0427f88 r4:c0e53968
[ 4667.414020] [<c0bdbcd4>] (dump_stack) from [<c02226c0>] (__warn+0xfc/0x114)
[ 4667.414029] [<c02225c4>] (__warn) from [<c0bd7c60>] (warn_slowpath_fmt+0x70/0xd8)
[ 4667.414037]  r7:0000152a r6:c0e53968 r5:c1205048 r4:00000000
[ 4667.414040] [<c0bd7bf4>] (warn_slowpath_fmt) from [<c0427f88>] (__alloc_pages+0x914/0x1138)
[ 4667.414050]  r9:0000000b r8:c043ed74 r7:00000cc0 r6:0041a000 r5:0000000b r4:00000000
[ 4667.414053] [<c0427674>] (__alloc_pages) from [<c03f6cc4>] (kmalloc_order+0x48/0xc0)
[ 4667.414063]  r10:0041a000 r9:0000000b r8:c043ed74 r7:00000cc0 r6:0041a000 r5:0000000b
[ 4667.414066]  r4:0041a000
[ 4667.414069] [<c03f6c7c>] (kmalloc_order) from [<c03f6d68>] (kmalloc_order_trace+0x2c/0xc4)
```

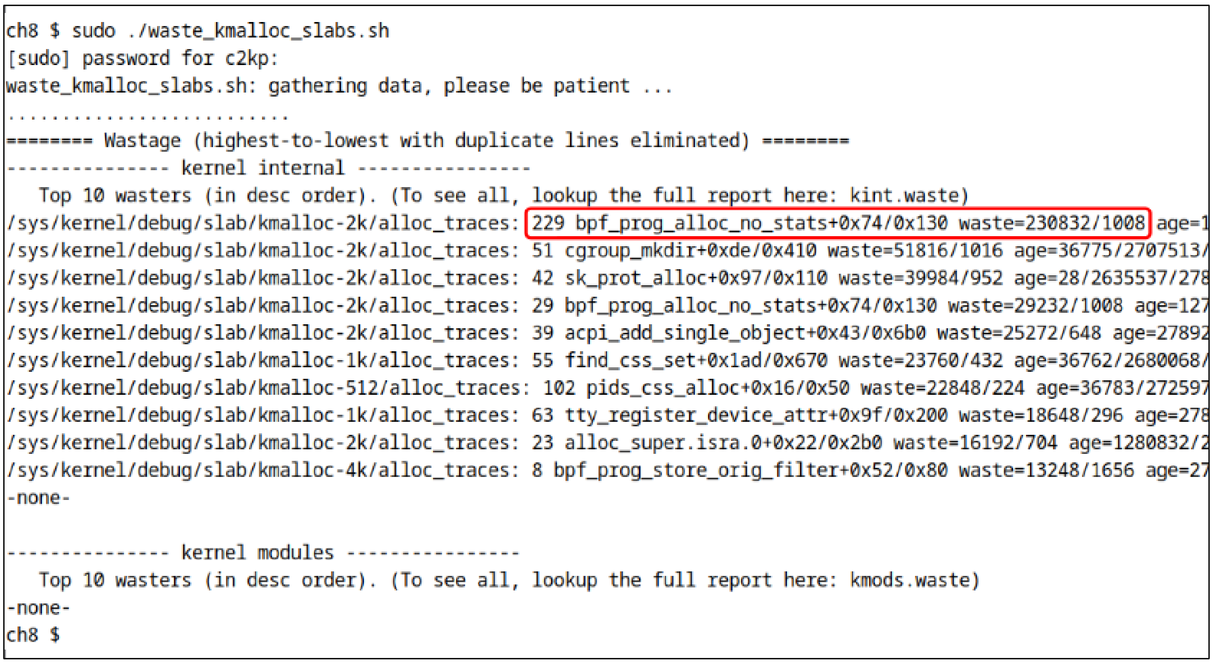


```
[ 4667.414053] [<c0427674>] (__alloc_pages) from [<c03f6cc4>] (kmalloc_order+0x48/0xc0)
[ 4667.414063]  r10:0041a000 r9:0000000b r8:c043ed74 r7:00000cc0 r6:0041a000 r5:0000000b
[ 4667.414066]  r4:0041a000
[ 4667.414069] [<c03f6c7c>] (kmalloc_order) from [<c03f6d68>] (kmalloc_order_trace+0x2c/0xc4)
[ 4667.414077]  r7:00000cc0 r6:0041a000 r5:c5800000 r4:0041a000
[ 4667.414080] [<c03f6d3c>] (kmalloc_order_trace) from [<c043ed74>] (__kmalloc+0x48c/0x4fc)
[ 4667.414091]  r10:0041a000 r9:00000cc0 r8:00000000 r7:bf318094 r6:bf319000 r5:c5800000
[ 4667.414094]  r4:0041a000
[ 4667.414097] [<c043e8e8>] (__kmalloc) from [<bf087048>] (slab3_maxsize_init+0x48/0x1000 [slab
3_maxsize])
[ 4667.414111]  r10:c1205048 r9:c3f52c48 r8:00000000 r7:bf318094 r6:bf319000 r5:c5800000
[ 4667.414114]  r4:0041a000
[ 4667.414116] [<bf087000>] (slab3_maxsize_init [slab3_maxsize]) from [<c02021a4>] (do_one_init
call+0x50/0x244)
[ 4667.414128]  r7:00000002 r6:bf087000 r5:c1205048 r4:bf319040
[ 4667.414131] [<c0202154>] (do_one_initcall) from [<c02d18e8>] (do_init_module+0x54/0x23c)
[ 4667.414141]  r8:bf319040 r7:00000002 r6:c3406f40 r5:00000002 r4:bf319040
[ 4667.414144] [<c02d1894>] (do_init_module) from [<c02d4148>] (load_module+0x24f8/0x294c)
[ 4667.414153]  r6:c3f52c00 r5:00000002 r4:c3f0df30
[ 4667.414156] [<c02d1c50>] (load_module) from [<c02d4818>] (sys_finit_module+0xc8/0xfc)
[ 4667.414166]  r10:0000017b r9:c3f0c000 r8:c0200244 r7:00000003 r6:0002de04 r5:00000000
[ 4667.414169]  r4:c1205048
[ 4667.414172] [<c02d4750>] (sys_finit_module) from [<c0200040>] (ret_fast_syscall+0x0/0x1c)
[ 4667.414180] Exception stack(0xc3f0dfa8 to 0xc3f0dff0)
[ 4667.414184] dfa0:                   00000000 00000002 00000003 0002de04 00000000 b6f28074
[ 4667.414189] dfc0: 00000000 00000002 9d2c7000 0000017b 01d98d68 00000002 be9eb7d4 00000000
[ 4667.414193] dfe0: be9eb600 be9eb5f0 00023bc0 b6c15580
[ 4667.414197]  r7:0000017b r6:9d2c7000 r5:00000002 r4:00000000
[ 4667.414201] ---[ end trace 439cc7506ad82102 ]---
[ 4667.414206] kmalloc fail, size2alloc=4300800
rpi4 $
```

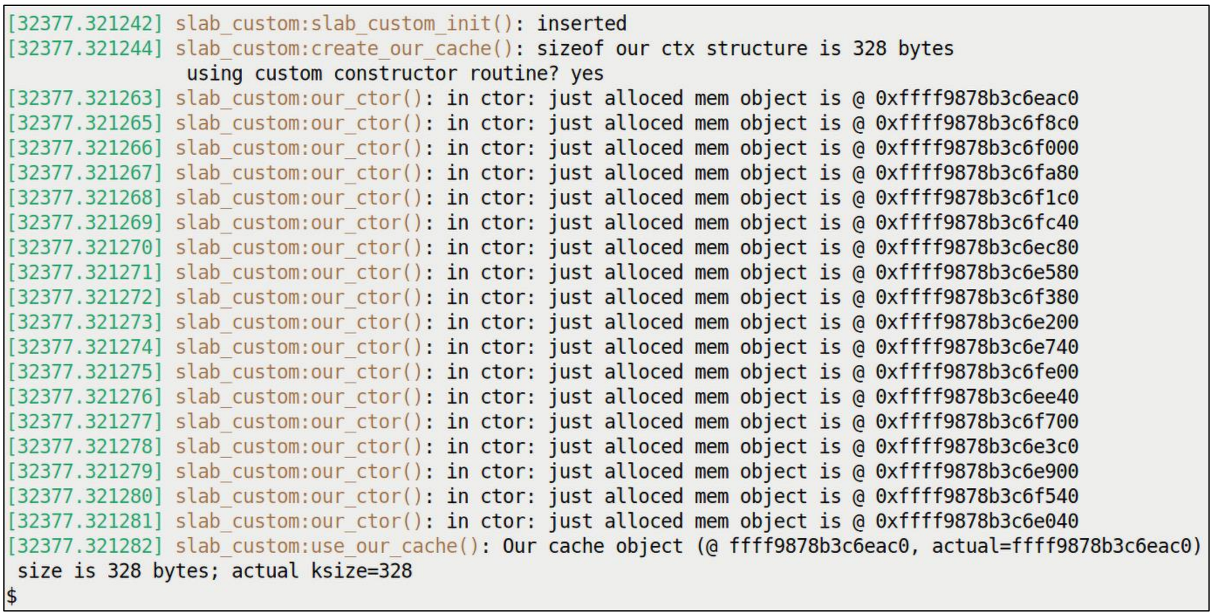


```
------------------------------
sudo insmod ./slab4_actualsize.ko && lsmod|grep slab4_actualsize
------------------------------
insmod: ERROR: could not insert module ./slab4_actualsize.ko: Cannot allocate memory
 ^--[FAILED]
------------------------------
sudo dmesg
------------------------------
[ 3948.215217] slab4_actualsize: inserted
[ 3948.215220] kmalloc(      n) :  Actual : Wastage : Waste %
[ 3948.215220] kmalloc(    100) :     128 :      28 :  28%
[ 3948.215232] kmalloc( 204900) :  262144 :   57244 :  27%
[ 3948.215249] kmalloc( 409700) :  524288 :  114588 :  27%
[ 3948.215513] kmalloc( 614500) : 1048576 :  434076 :  70%
[ 3948.215537] kmalloc( 819300) : 1048576 :  229276 :  27%
[ 3948.215559] kmalloc(1024100) : 1048576 :   24476 :   2%
[ 3948.215617] kmalloc(1228900) : 2097152 :  868252 :  70%
[ 3948.215660] kmalloc(1433700) : 2097152 :  663452 :  46%
[ 3948.215700] kmalloc(1638500) : 2097152 :  458652 :  27%
[ 3948.215741] kmalloc(1843300) : 2097152 :  253852 :  13%
[ 3948.215782] kmalloc(2048100) : 2097152 :   49052 :   2%
[ 3948.216309] kmalloc(2252900) : 4194304 : 1941404 :  86%
[ 3948.216396] kmalloc(2457700) : 4194304 : 1736604 :  70%
[ 3948.216478] kmalloc(2662500) : 4194304 : 1531804 :  57%
[ 3948.216614] kmalloc(2867300) : 4194304 : 1327004 :  46%
[ 3948.216710] kmalloc(3072100) : 4194304 : 1122204 :  36%
[ 3948.216792] kmalloc(3276900) : 4194304 :  917404 :  27%
[ 3948.216874] kmalloc(3481700) : 4194304 :  712604 :  20%
[ 3948.216956] kmalloc(3686500) : 4194304 :  507804 :  13%
[ 3948.217038] kmalloc(3891300) : 4194304 :  303004 :   7%
[ 3948.217120] kmalloc(4096100) : 4194304 :   98204 :   2%
[ 3948.217126] ------------[ cut here ]------------
[ 3948.217127] WARNING: CPU: 2 PID: 124052 at mm/page_alloc.c:5534 __alloc_pages+0x2
2a/0x1270
[ 3948.217135] Modules linked in: slab4_actualsize(OE+) tls drm_ttm_helper ttm drm_k
ms_helper syscopyarea sysfillrect sysimgblt fb_sys_fops vboxsf(OE) binfmt_misc snd_i
ntel8x0 snd_ac97_codec ac97_bus snd_pcm intel_rapl_msr snd_seq joydev snd_timer snd_
seq_device intel_rapl_common crct10dif_pclmul crc32_pclmul ghash_clmulni_intel aesni
_intel snd crypto_simd input_leds cryptd video rapl wmi serio_raw soundcore vboxgues
t(OE) mac_hid drm sch_fq_codel msr parport_pc ppdev lp parport ramoops pstore_blk re
ed_solomon efi_pstore pstore_zone ip_tables x_tables autofs4 hid_generic usbhid hid
psmouse e1000 ahci i2c_piix4 libahci pata_acpi
[ 3948.217165] CPU: 2 PID: 124052 Comm: insmod Tainted: G           OE      6.1.11-l
kp-kernel #1
[ 3948.217167] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12
/01/2006
[ 3948.217168] RIP: 0010:__alloc_pages+0x22a/0x1270
```

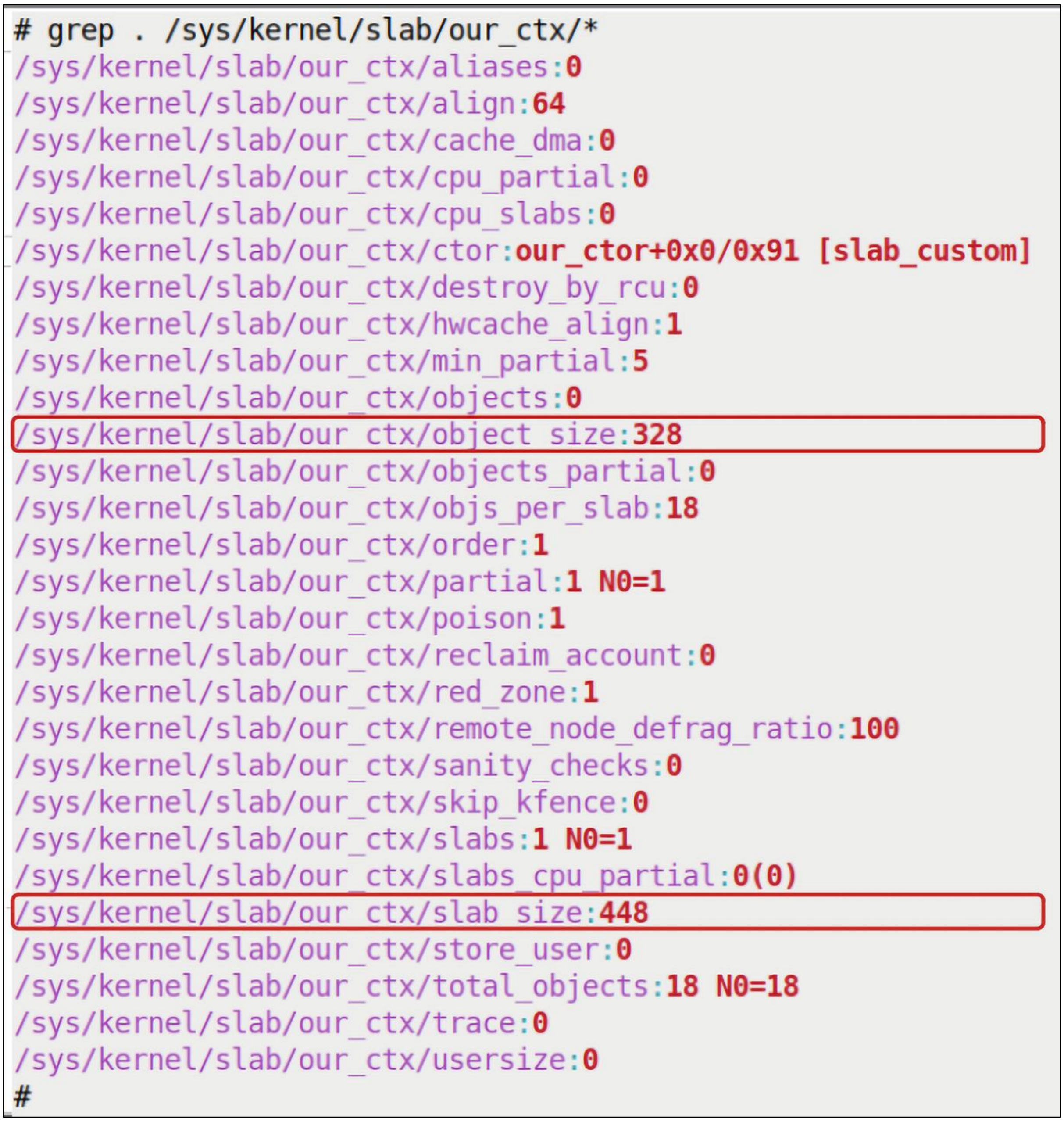




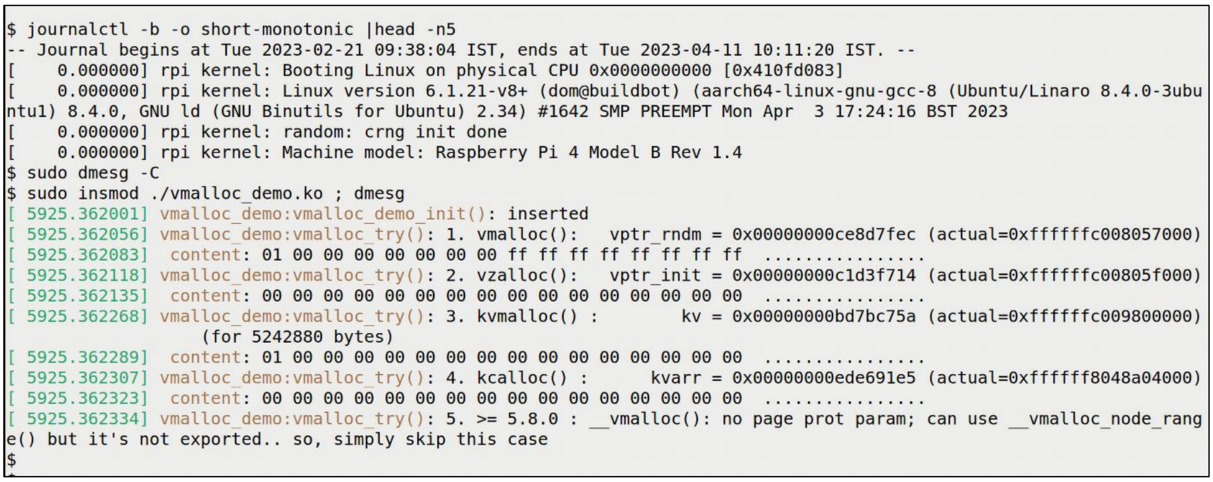
```
ch8 $ sudo ./waste_kmalloc_slabs.sh
[sudo] password for c2kp:
waste_kmalloc_slabs.sh: gathering data, please be patient ...
..........................
======== Wastage (highest-to-lowest with duplicate lines eliminated) ========
--------------- kernel internal ----------------
   Top 10 wasters (in desc order). (To see all, lookup the full report here: kint.waste)
/sys/kernel/debug/slab/kmalloc-2k/alloc_traces: 229 bpf_prog_alloc_no_stats+0x74/0x130 waste=230832/1008 age=1
/sys/kernel/debug/slab/kmalloc-2k/alloc_traces: 51 cgroup_mkdir+0xde/0x410 waste=51816/1016 age=36775/2707513/
/sys/kernel/debug/slab/kmalloc-2k/alloc_traces: 42 sk_prot_alloc+0x97/0x110 waste=39984/952 age=28/2635537/278
/sys/kernel/debug/slab/kmalloc-2k/alloc_traces: 29 bpf_prog_alloc_no_stats+0x74/0x130 waste=29232/1008 age=127
/sys/kernel/debug/slab/kmalloc-2k/alloc_traces: 39 acpi_add_single_object+0x43/0x6b0 waste=25272/648 age=27892
/sys/kernel/debug/slab/kmalloc-1k/alloc_traces: 55 find_css_set+0x1ad/0x670 waste=23760/432 age=36762/2680068/
/sys/kernel/debug/slab/kmalloc-512/alloc_traces: 102 pids_css_alloc+0x16/0x50 waste=22848/224 age=36783/272597
/sys/kernel/debug/slab/kmalloc-1k/alloc_traces: 63 tty_register_device_attr+0x9f/0x200 waste=18648/296 age=278
/sys/kernel/debug/slab/kmalloc-2k/alloc_traces: 23 alloc_super.isra.0+0x22/0x2b0 waste=16192/704 age=1280832/2
/sys/kernel/debug/slab/kmalloc-4k/alloc_traces: 8 bpf_prog_store_orig_filter+0x52/0x80 waste=13248/1656 age=27
-none-

--------------- kernel modules ----------------
   Top 10 wasters (in desc order). (To see all, lookup the full report here: kmods.waste)
-none-
ch8 $
```



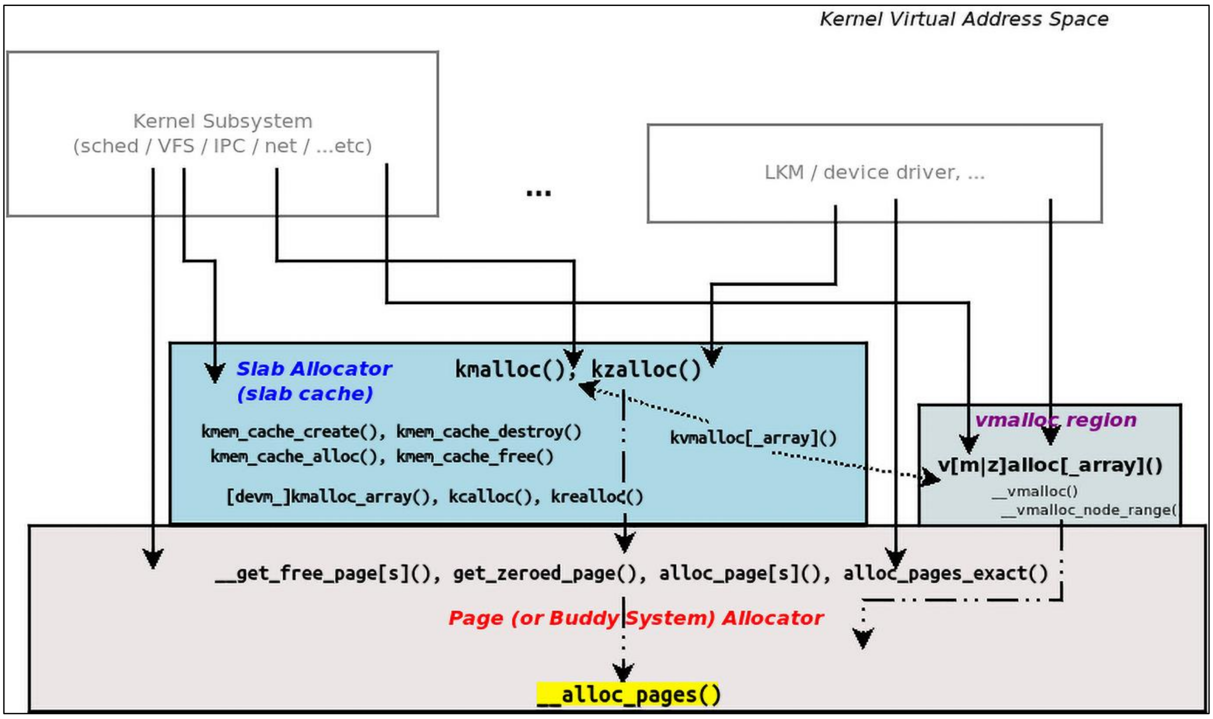
```
[32377.321242] slab_custom:slab_custom_init(): inserted
[32377.321244] slab_custom:create_our_cache(): sizeof our ctx structure is 328 bytes
                using custom constructor routine? yes
[32377.321263] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6eac0
[32377.321265] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6f8c0
[32377.321266] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6f000
[32377.321267] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6fa80
[32377.321268] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6f1c0
[32377.321269] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6fc40
[32377.321270] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6ec80
[32377.321271] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6e580
[32377.321272] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6f380
[32377.321273] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6e200
[32377.321274] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6e740
[32377.321275] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6fe00
[32377.321276] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6ee40
[32377.321277] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6f700
[32377.321278] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6e3c0
[32377.321279] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6e900
[32377.321280] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6f540
[32377.321281] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff9878b3c6e040
[32377.321282] slab_custom:use_our_cache(): Our cache object (@ ffff9878b3c6eac0, actual=ffff9878b3c6eac0)
 size is 328 bytes; actual ksize=328
$
```

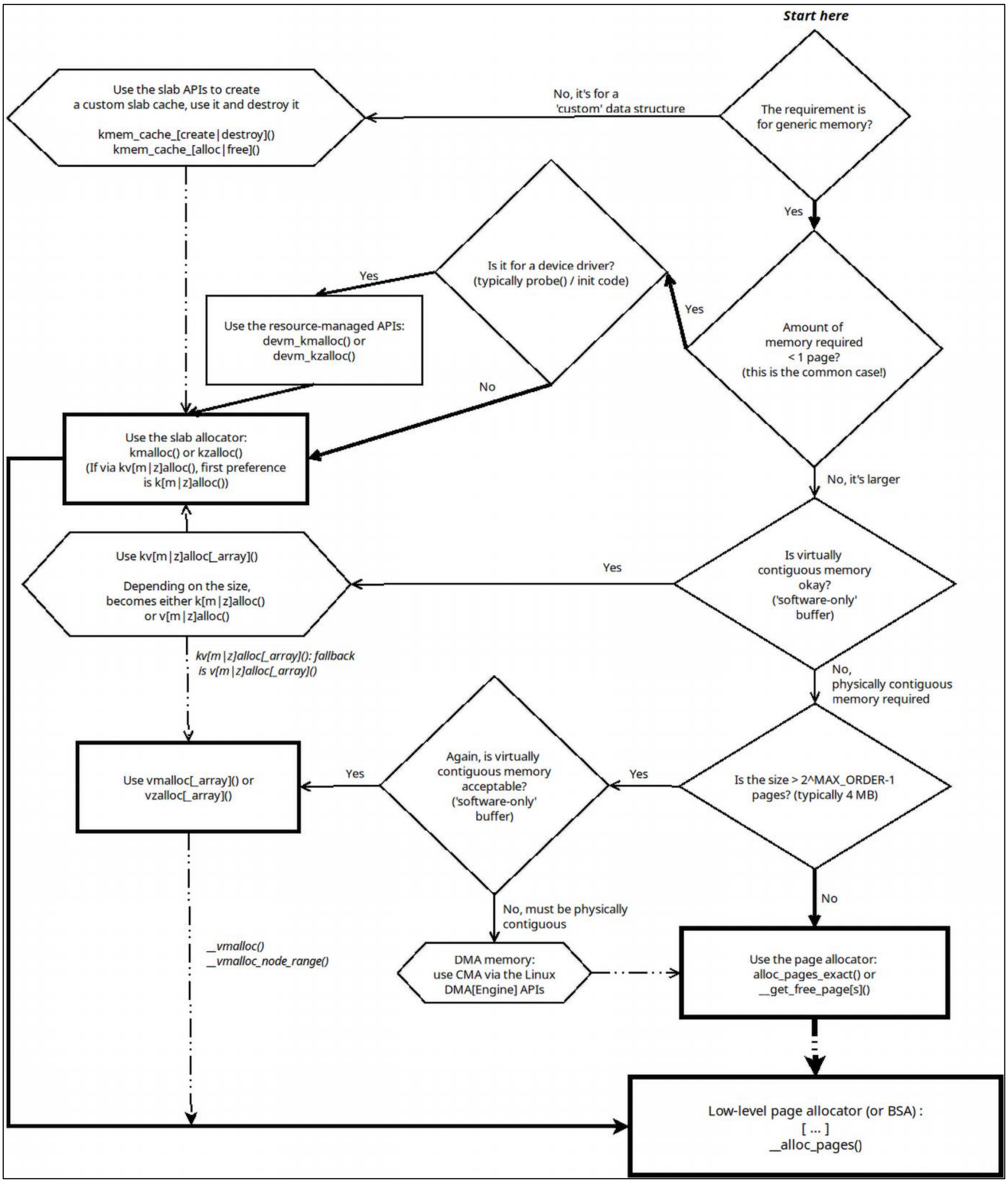


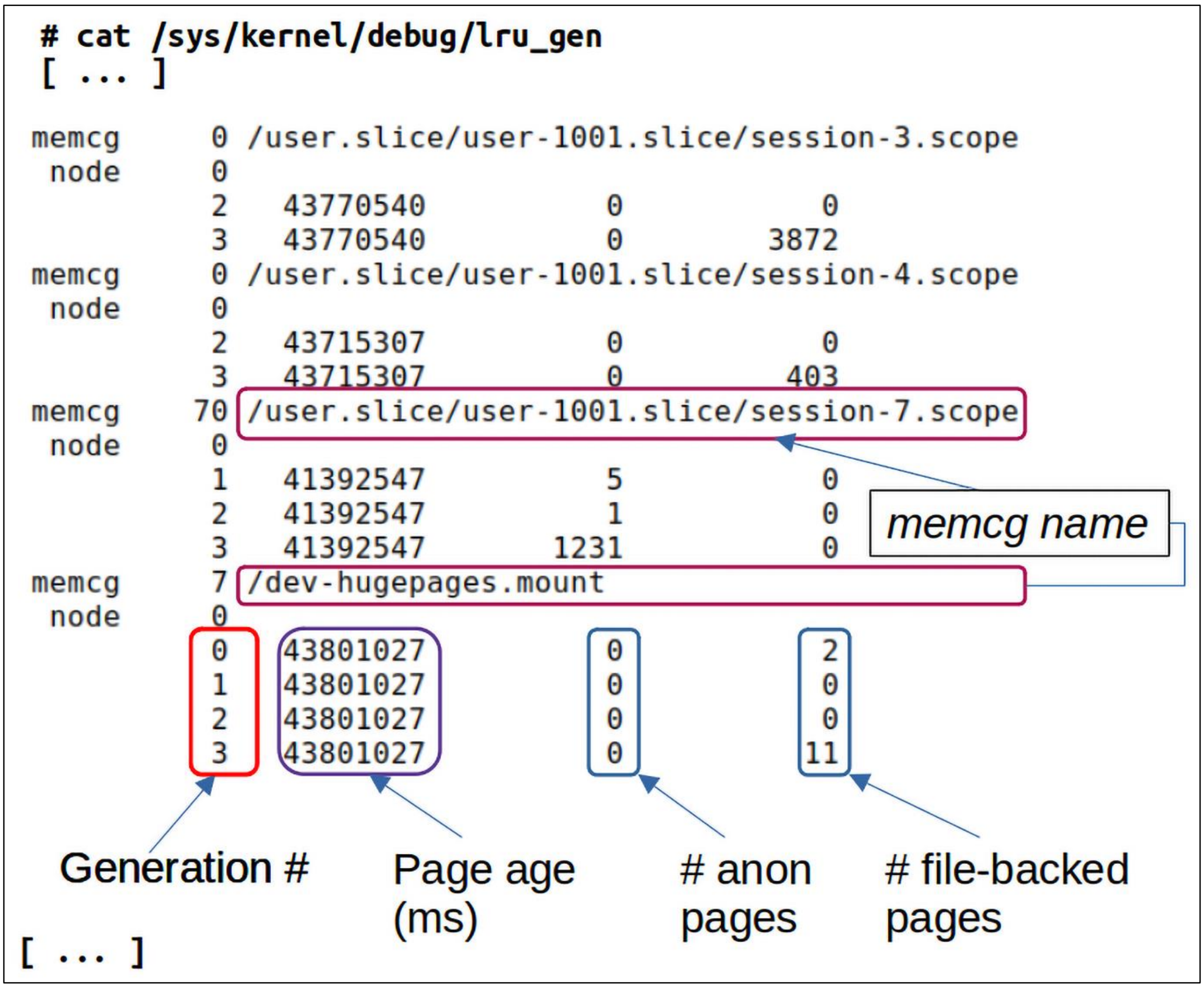
```
# grep . /sys/kernel/slab/our_ctx/*
/sys/kernel/slab/our_ctx/aliases:0
/sys/kernel/slab/our_ctx/align:64
/sys/kernel/slab/our_ctx/cache_dma:0
/sys/kernel/slab/our_ctx/cpu_partial:0
/sys/kernel/slab/our_ctx/cpu_slabs:0
/sys/kernel/slab/our_ctx/ctor:our_ctor+0x0/0x91 [slab_custom]
/sys/kernel/slab/our_ctx/destroy_by_rcu:0
/sys/kernel/slab/our_ctx/hwcache_align:1
/sys/kernel/slab/our_ctx/min_partial:5
/sys/kernel/slab/our_ctx/objects:0
/sys/kernel/slab/our_ctx/object_size:328
/sys/kernel/slab/our_ctx/objects_partial:0
/sys/kernel/slab/our_ctx/objs_per_slab:18
/sys/kernel/slab/our_ctx/order:1
/sys/kernel/slab/our_ctx/partial:1 N0=1
/sys/kernel/slab/our_ctx/poison:1
/sys/kernel/slab/our_ctx/reclaim_account:0
/sys/kernel/slab/our_ctx/red_zone:1
/sys/kernel/slab/our_ctx/remote_node_defrag_ratio:100
/sys/kernel/slab/our_ctx/sanity_checks:0
/sys/kernel/slab/our_ctx/skip_kfence:0
/sys/kernel/slab/our_ctx/slabs:1 N0=1
/sys/kernel/slab/our_ctx/slabs_cpu_partial:0(0)
/sys/kernel/slab/our_ctx/slab_size:448
/sys/kernel/slab/our_ctx/store_user:0
/sys/kernel/slab/our_ctx/total_objects:18 N0=18
/sys/kernel/slab/our_ctx/trace:0
/sys/kernel/slab/our_ctx/usersize:0
#
```



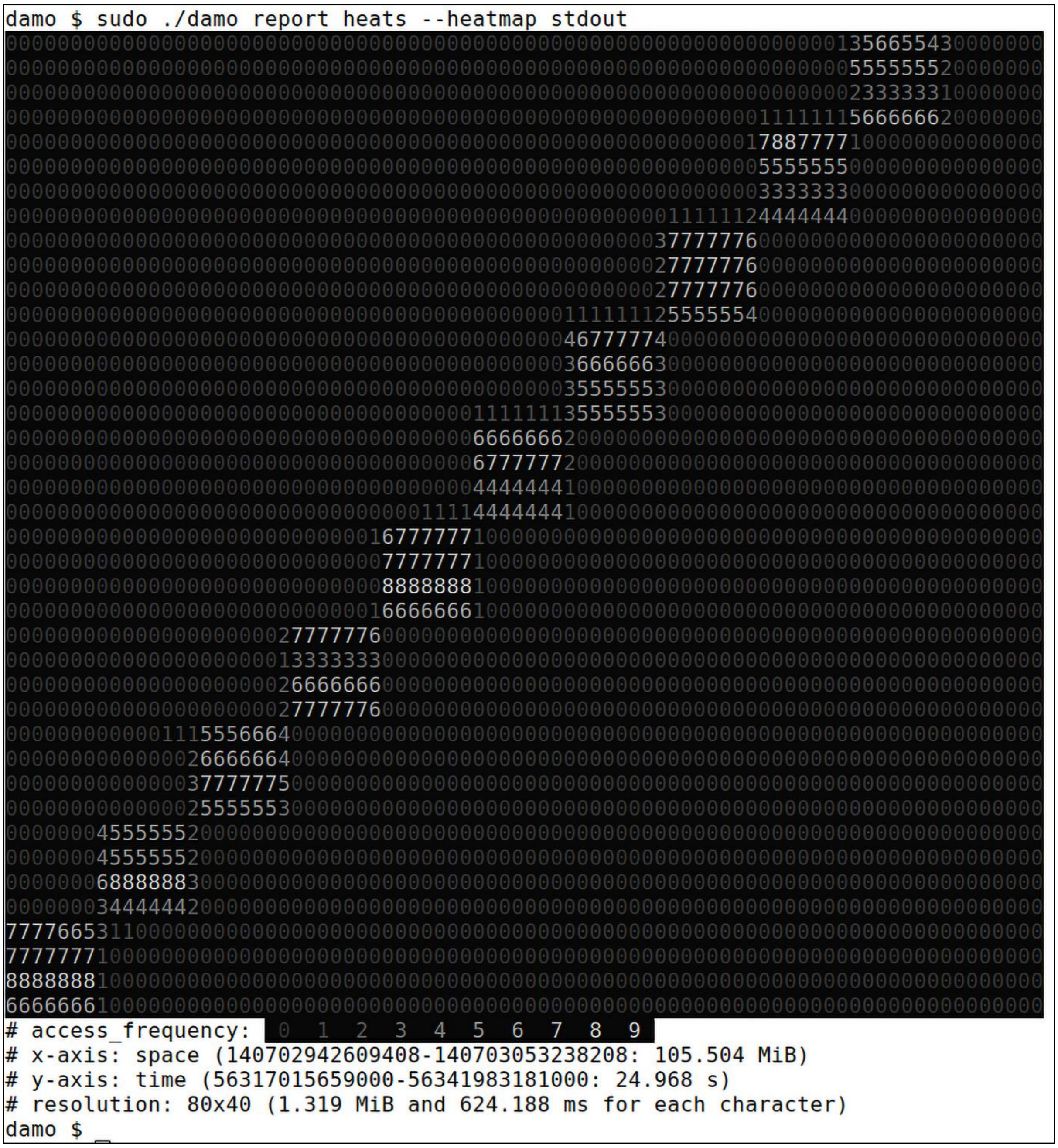
```
$ journalctl -b -o short-monotonic |head -n5
-- Journal begins at Tue 2023-02-21 09:38:04 IST, ends at Tue 2023-04-11 10:11:20 IST. --
[    0.000000] rpi kernel: Booting Linux on physical CPU 0x0000000000 [0x410fd083]
[    0.000000] rpi kernel: Linux version 6.1.21-v8+ (dom@buildbot) (aarch64-linux-gnu-gcc-8 (Ubuntu/Linaro 8.4.0-3ubu
ntu1) 8.4.0, GNU ld (GNU Binutils for Ubuntu) 2.34) #1642 SMP PREEMPT Mon Apr  3 17:24:16 BST 2023
[    0.000000] rpi kernel: random: crng init done
[    0.000000] rpi kernel: Machine model: Raspberry Pi 4 Model B Rev 1.4
$ sudo dmesg -C
$ sudo insmod ./vmalloc_demo.ko ; dmesg
[ 5925.362001] vmalloc_demo:vmalloc_demo_init(): inserted
[ 5925.362056] vmalloc_demo:vmalloc_try(): 1. vmalloc():   vptr_rndm = 0x00000000ce8d7fec (actual=0xffffffc008057000)
[ 5925.362083]  content: 01 00 00 00 00 00 00 00 ff ff ff ff ff ff ff ff  ................
[ 5925.362118] vmalloc_demo:vmalloc_try(): 2. vzalloc():   vptr_init = 0x00000000c1d3f714 (actual=0xffffffc00805f000)
[ 5925.362135]  content: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[ 5925.362268] vmalloc_demo:vmalloc_try(): 3. kvmalloc() :        kv = 0x00000000bd7bc75a (actual=0xffffffc009800000)
                  (for 5242880 bytes)
[ 5925.362289]  content: 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[ 5925.362307] vmalloc_demo:vmalloc_try(): 4. kcalloc() :      kvarr = 0x00000000ede691e5 (actual=0xffffff8048a04000)
[ 5925.362323]  content: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[ 5925.362334] vmalloc_demo:vmalloc_try(): 5. >= 5.8.0 : __vmalloc(): no page prot param; can use __vmalloc_node_rang
e() but it's not exported.. so, simply skip this case
$
```


Kernel Virtual Address Space
Kernel Subsystem
(sched / VFS / IPC / net / ...etc)
...
LKM / device driver, ...
Slab Allocator
(slab cache)
kmalloc(), kzalloc()
kmem_cache_create(), kmem_cache_destroy()
kmem_cache_alloc(), kmem_cache_free()
kvmalloc[_array]()
[devm_]kmalloc_array(), kcalloc(), krealloc()
vmalloc region
v[m|z]alloc[_array]()
__vmalloc()
__vmalloc_node_range(
__get_free_page[s](), get_zeroed_page(), alloc_page[s](), alloc_pages_exact()
Page (or Buddy System) Allocator
__alloc_pages()

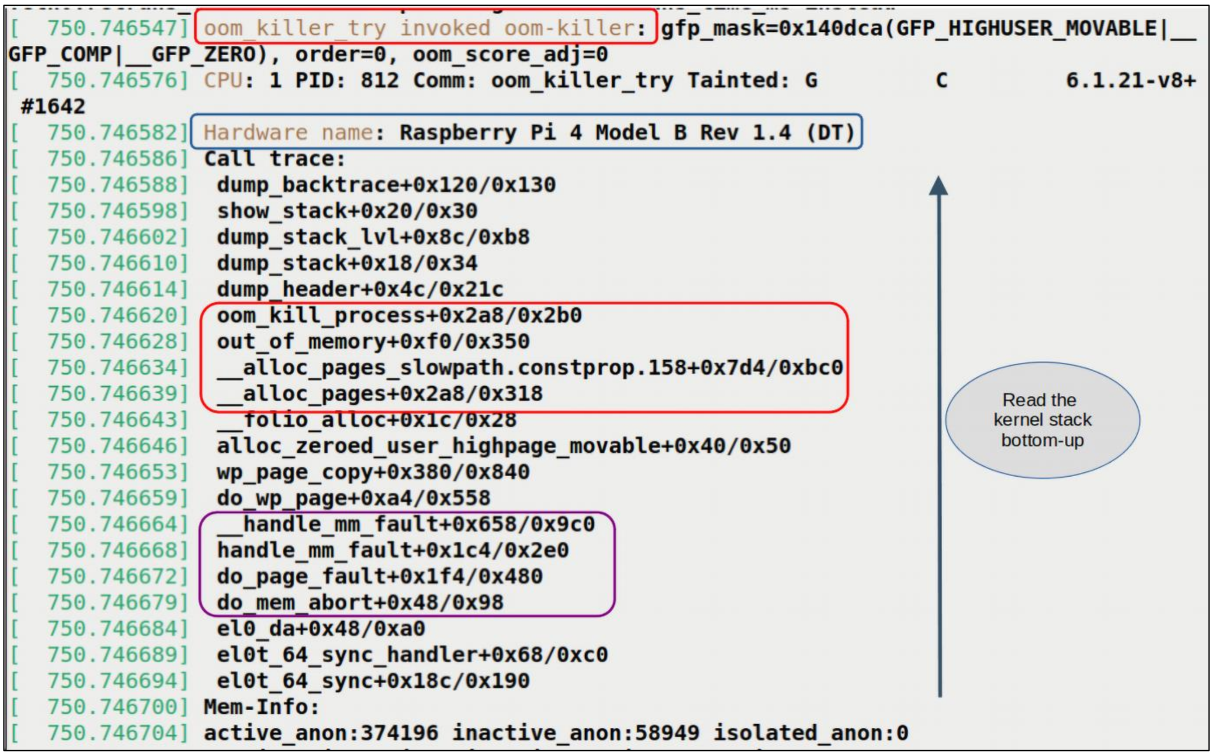

Start here
The requirement is for generic memory?
No, it's for a 'custom' data structure
Use the slab APIs to create a custom slab cache, use it and destroy it
kmem_cache_[create|destroy]()
kmem_cache_[alloc|free]()
Yes
Amount of memory required < 1 page? (this is the common case!)
Yes
Is it for a device driver? (typically probe() / init code)
Yes
Use the resource-managed APIs: devm_kmalloc() or devm_kzalloc()
No
Use the slab allocator: kmalloc() or kzalloc() (If via kv[m|z]alloc(), first preference is k[m|z]alloc())
No, it's larger
Is virtually contiguous memory okay? ('software-only' buffer)
Yes
Use kv[m|z]alloc[_array]()
Depending on the size, becomes either k[m|z]alloc() or v[m|z]alloc()
kv[m|z]alloc[_array](): fallback is v[m|z]alloc[_array]()
No, physically contiguous memory required
Is the size > 2^MAX_ORDER-1 pages? (typically 4 MB)
Yes
Again, is virtually contiguous memory acceptable? ('software-only' buffer)
Yes
Use vmalloc[_array]() or vzalloc[_array]()
__vmalloc()
__vmalloc_node_range()
No, must be physically contiguous
DMA memory: use CMA via the Linux DMA[Engine] APIs
No
Use the page allocator: alloc_pages_exact() or __get_free_page[s]()
Low-level page allocator (or BSA) :
[...]
__alloc_pages()



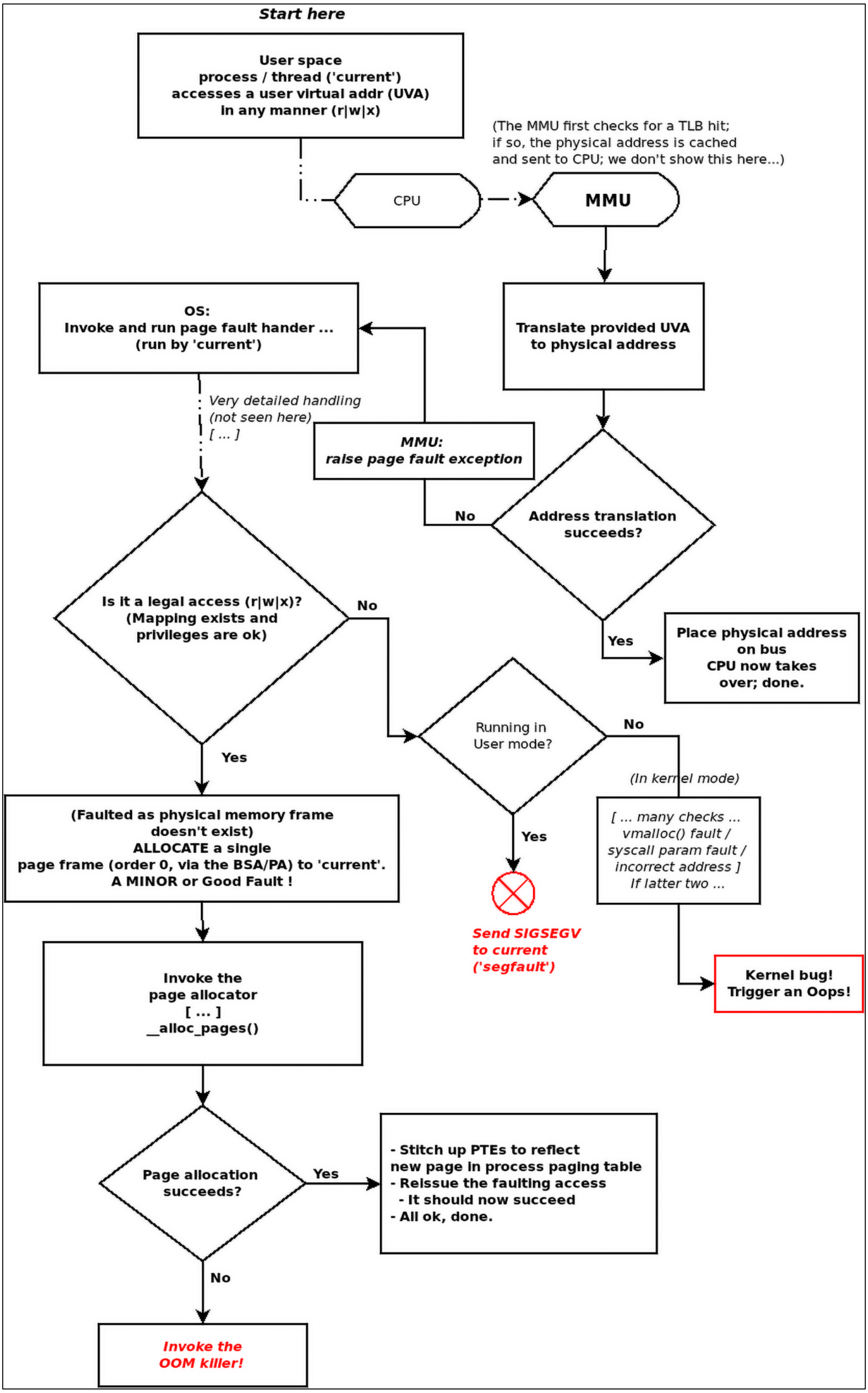
```
 # cat /sys/kernel/debug/lru_gen
 [ ... ]

memcg     0 /user.slice/user-1001.slice/session-3.scope
 node     0
          2   43770540          0           0
          3   43770540          0        3872
memcg     0 /user.slice/user-1001.slice/session-4.scope
 node     0
          2   43715307          0           0
          3   43715307          0         403
memcg    70 /user.slice/user-1001.slice/session-7.scope
 node     0
          1   41392547          5           0
          2   41392547          1           0
          3   41392547       1231           0
memcg     7 /dev-hugepages.mount
 node     0
          0   43801027          0           2
          1   43801027          0           0
          2   43801027          0           0
          3   43801027          0          11

[ ... ]
```

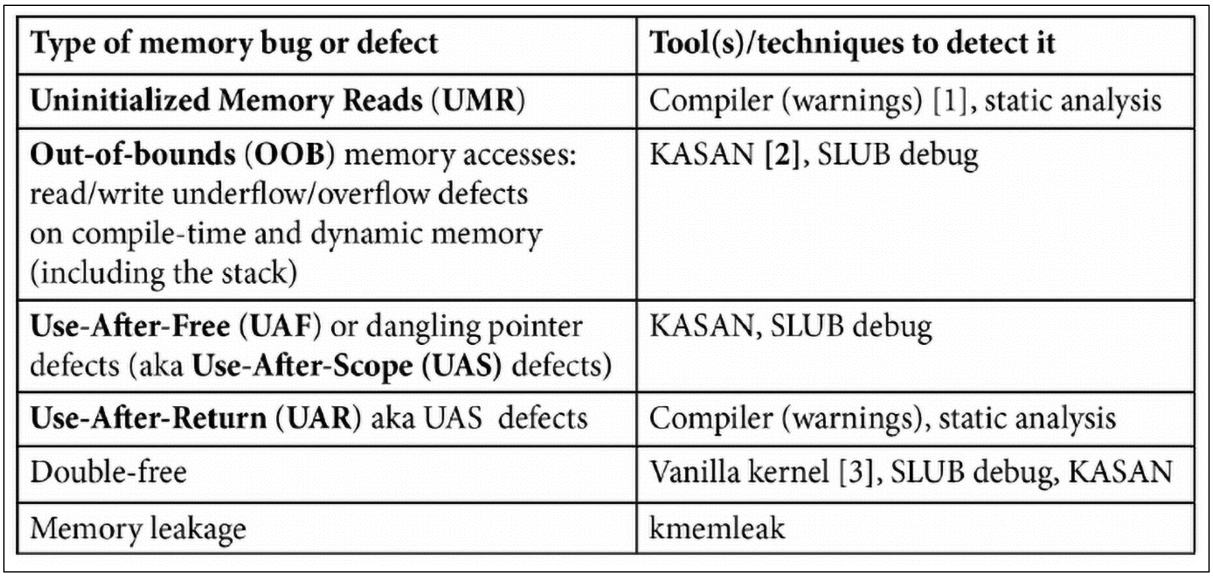


```
damo $ sudo ./damo report heats --heatmap stdout
00000000000000000000000000000000000000000000000000000000000000001356655430000000
00000000000000000000000000000000000000000000000000000000000000000555555520000000
00000000000000000000000000000000000000000000000000000000000000000233333310000000
00000000000000000000000000000000000000000000000000000000001111111566666620000000
00000000000000000000000000000000000000000000000000000000017887777100000000000000
00000000000000000000000000000000000000000000000000000000005555555000000000000000
00000000000000000000000000000000000000000000000000000000003333333000000000000000
00000000000000000000000000000000000000000000000000011111124444444000000000000000
00000000000000000000000000000000000000000000000000377777760000000000000000000000
00000000000000000000000000000000000000000000000000277777760000000000000000000000
00000000000000000000000000000000000000000000000000277777760000000000000000000000
00000000000000000000000000000000000000000001111111255555540000000000000000000000
00000000000000000000000000000000000000000004677777400000000000000000000000000000
00000000000000000000000000000000000000000003666666300000000000000000000000000000
00000000000000000000000000000000000000000003555555300000000000000000000000000000
00000000000000000000000000000000000011111113555555300000000000000000000000000000
00000000000000000000000000000000000066666662000000000000000000000000000000000000
00000000000000000000000000000000000067777772000000000000000000000000000000000000
00000000000000000000000000000000000044444441000000000000000000000000000000000000
00000000000000000000000000000000111144444441000000000000000000000000000000000000
00000000000000000000000000001677777710000000000000000000000000000000000000000000
00000000000000000000000000000777777710000000000000000000000000000000000000000000
00000000000000000000000000000888888810000000000000000000000000000000000000000000
00000000000000000000000000001666666610000000000000000000000000000000000000000000
00000000000000000000027777776000000000000000000000000000000000000000000000000000
00000000000000000000013333333000000000000000000000000000000000000000000000000000
00000000000000000000026666666000000000000000000000000000000000000000000000000000
00000000000000000000027777776000000000000000000000000000000000000000000000000000
00000000000011155566640000000000000000000000000000000000000000000000000000000000
00000000000000266666640000000000000000000000000000000000000000000000000000000000
00000000000000377777750000000000000000000000000000000000000000000000000000000000
00000000000000255555530000000000000000000000000000000000000000000000000000000000
00000004555555200000000000000000000000000000000000000000000000000000000000000000
00000004555555200000000000000000000000000000000000000000000000000000000000000000
00000006888888300000000000000000000000000000000000000000000000000000000000000000
00000003444444200000000000000000000000000000000000000000000000000000000000000000
77776653110000000000000000000000000000000000000000000000000000000000000000000000
77777771000000000000000000000000000000000000000000000000000000000000000000000000
88888881000000000000000000000000000000000000000000000000000000000000000000000000
66666661000000000000000000000000000000000000000000000000000000000000000000000000
# access_frequency:  0  1  2  3  4  5  6  7  8  9
# x-axis: space (140702942609408-140703053238208: 105.504 MiB)
# y-axis: time (56317015659000-56341983181000: 24.968 s)
# resolution: 80x40 (1.319 MiB and 624.188 ms for each character)
damo $ _
```

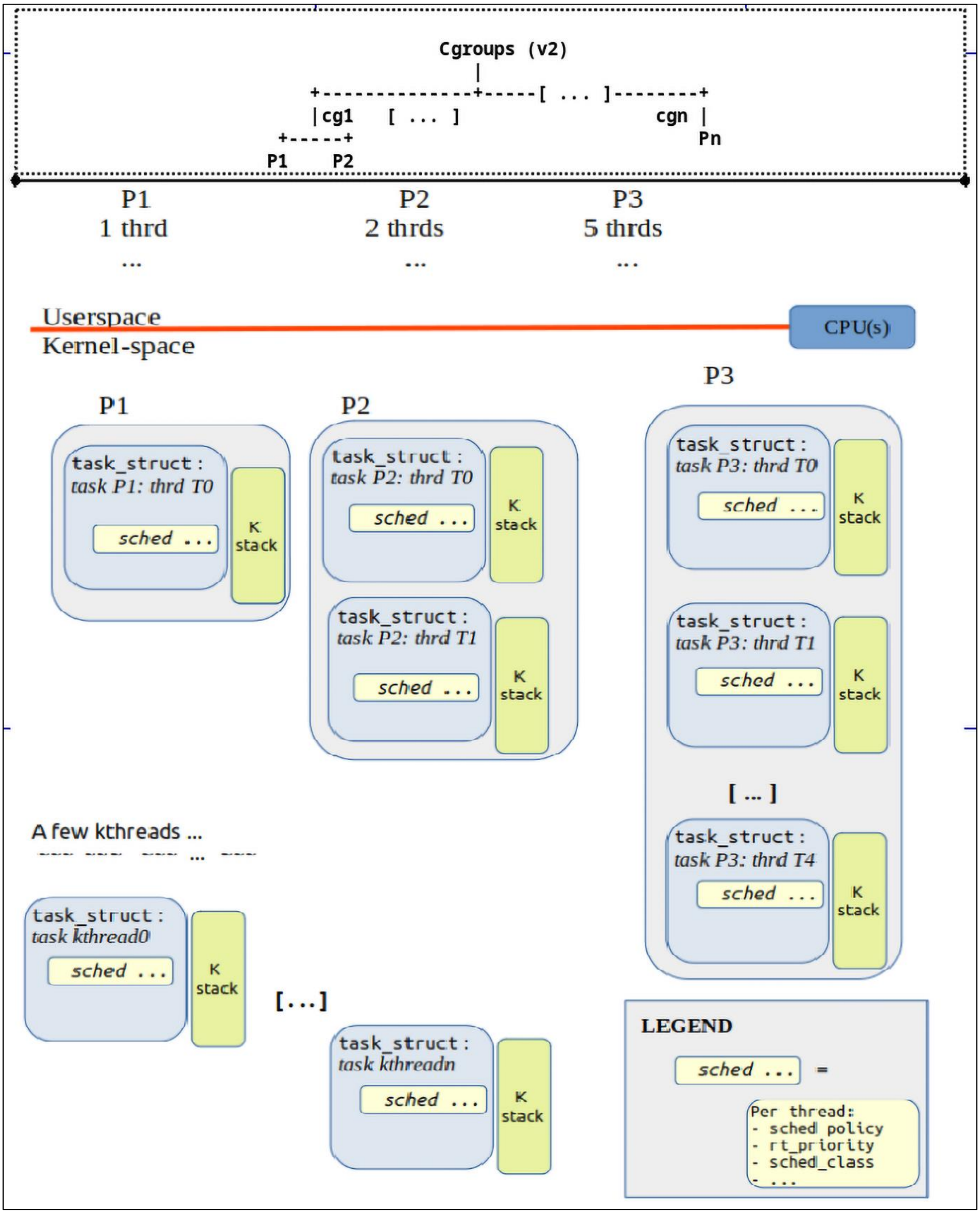


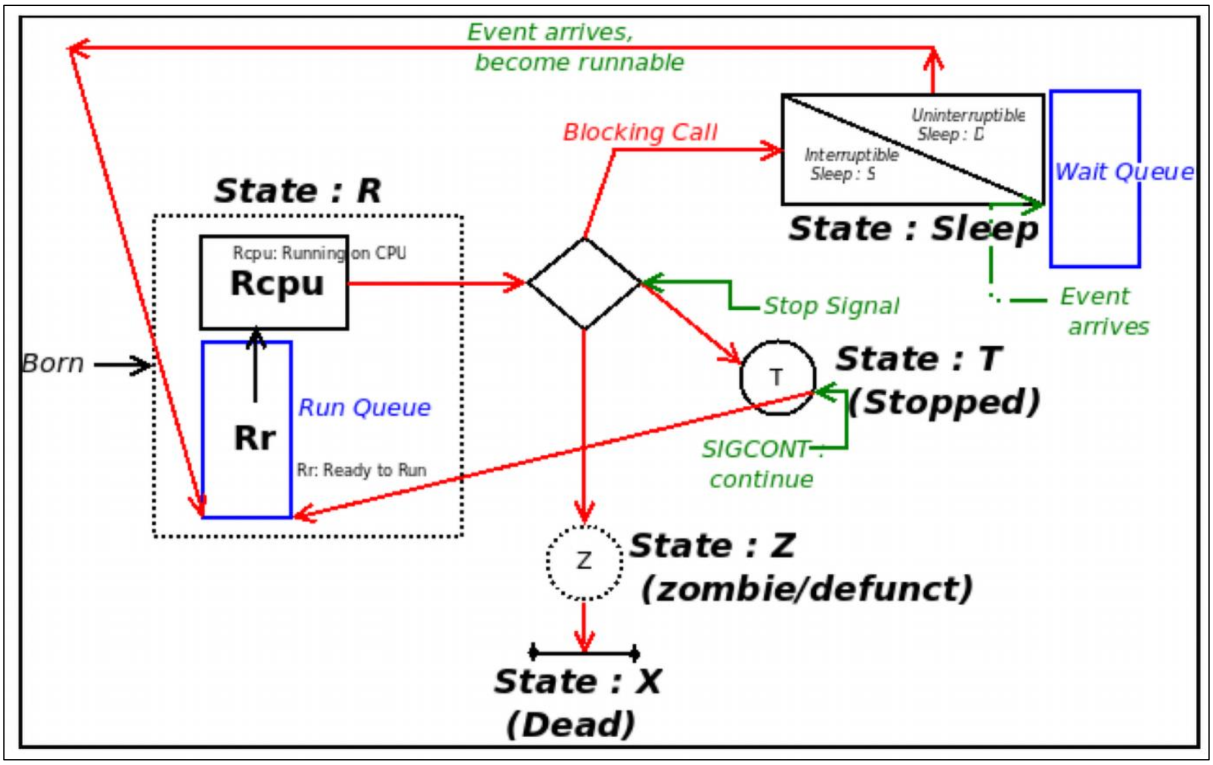
```
[  750.746547] oom_killer_try invoked oom-killer: gfp_mask=0x140dca(GFP_HIGHUSER_MOVABLE|__
GFP_COMP|__GFP_ZERO), order=0, oom_score_adj=0
[  750.746576] CPU: 1 PID: 812 Comm: oom_killer_try Tainted: G         C         6.1.21-v8+
 #1642
[  750.746582] Hardware name: Raspberry Pi 4 Model B Rev 1.4 (DT)
[  750.746586] Call trace:
[  750.746588]  dump_backtrace+0x120/0x130
[  750.746598]  show_stack+0x20/0x30
[  750.746602]  dump_stack_lvl+0x8c/0xb8
[  750.746610]  dump_stack+0x18/0x34
[  750.746614]  dump_header+0x4c/0x21c
[  750.746620]  oom_kill_process+0x2a8/0x2b0
[  750.746628]  out_of_memory+0xf0/0x350
[  750.746634]  __alloc_pages_slowpath.constprop.158+0x7d4/0xbc0
[  750.746639]  __alloc_pages+0x2a8/0x318
[  750.746643]  __folio_alloc+0x1c/0x28
[  750.746646]  alloc_zeroed_user_highpage_movable+0x40/0x50
[  750.746653]  wp_page_copy+0x380/0x840
[  750.746659]  do_wp_page+0xa4/0x558
[  750.746664]  __handle_mm_fault+0x658/0x9c0
[  750.746668]  handle_mm_fault+0x1c4/0x2e0
[  750.746672]  do_page_fault+0x1f4/0x480
[  750.746679]  do_mem_abort+0x48/0x98
[  750.746684]  el0_da+0x48/0xa0
[  750.746689]  el0t_64_sync_handler+0x68/0xc0
[  750.746694]  el0t_64_sync+0x18c/0x190
[  750.746700] Mem-Info:
[  750.746704] active_anon:374196 inactive_anon:58949 isolated_anon:0
```

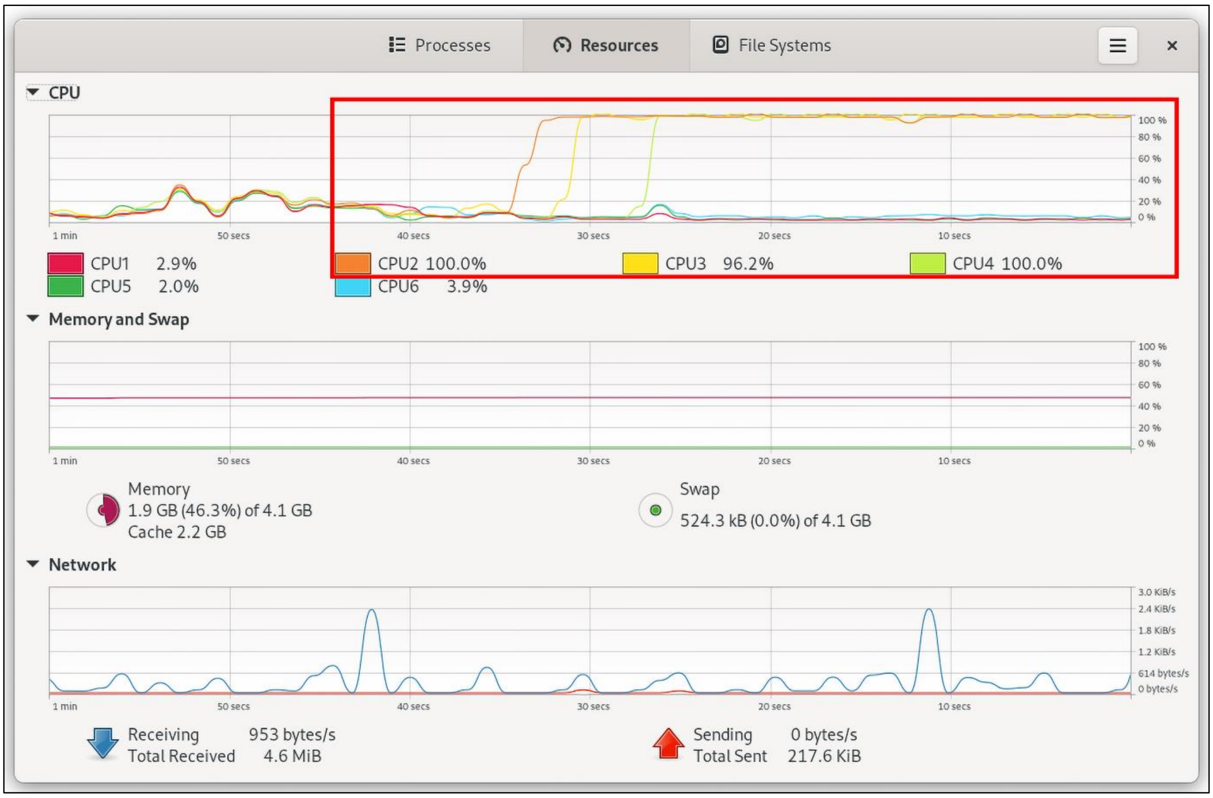

Start here
User space
process / thread ('current')
accesses a user virtual addr (UVA)
in any manner (r|w|x)
(The MMU first checks for a TLB hit;
if so, the physical address is cached
and sent to CPU; we don't show this here...)
CPU
MMU
OS:
Invoke and run page fault hander ...
(run by 'current')
Translate provided UVA
to physical address
Very detailed handling
(not seen here)
[...]
MMU:
raise page fault exception
No
Address translation
succeeds?
Is it a legal access (r|w|x)?
(Mapping exists and
privileges are ok)
No
Yes
Place physical address
on bus
CPU now takes
over; done.
Running in
User mode?
No
(In kernel mode)
Yes
(Faulted as physical memory frame
doesn't exist)
ALLOCATE a single
page frame (order 0, via the BSA/PA) to 'current'.
A MINOR or Good Fault !
Yes
[... many checks ...
vmalloc() fault /
syscall param fault /
incorrect address]
If latter two ...
Send SIGSEGV
to current
('segfault')
Kernel bug!
Trigger an Oops!
Invoke the
page allocator
[...]
__alloc_pages()
Page allocation
succeeds?
Yes
- Stitch up PTEs to reflect
new page in process paging table
- Reissue the faulting access
- It should now succeed
- All ok, done.
No
Invoke the
OOM killer!



Type of memory bug or defect	**Tool(s)/techniques to detect it**
Uninitialized Memory Reads (**UMR**)	Compiler (warnings) [1], static analysis
Out-of-bounds (**OOB**) memory accesses: read/write underflow/overflow defects on compile-time and dynamic memory (including the stack)	KASAN **[2]**, SLUB debug
Use-After-Free (**UAF**) or dangling pointer defects (aka **Use-After-Scope (UAS)** defects)	KASAN, SLUB debug
Use-After-Return (**UAR**) aka UAS defects	Compiler (warnings), static analysis
Double-free	Vanilla kernel [3], SLUB debug, KASAN
Memory leakage	kmemleak


Cgroups (v2)
cg1
[...]
cgn
P1
P2
Pn
P1
1 thrd
...
P2
2 thrds
...
P3
5 thrds
...
Userspace
Kernel-space
CPU(s)
P1
task_struct :
task P1: thrd T0
sched ...
K stack
P2
task_struct :
task P2: thrd T0
sched ...
K stack
task_struct :
task P2: thrd T1
sched ...
K stack
P3
task_struct :
task P3: thrd T0
sched ...
K stack
task_struct :
task P3: thrd T1
sched ...
K stack
[...]
task_struct :
task P3: thrd T4
sched ...
K stack
A few kthreads ...
task_struct :
task kthread0
sched ...
K stack
[...]
task_struct :
task kthreadn
sched ...
K stack
LEGEND
sched ... =
Per thread:
- sched policy
- rt_priority
- sched_class
- ...





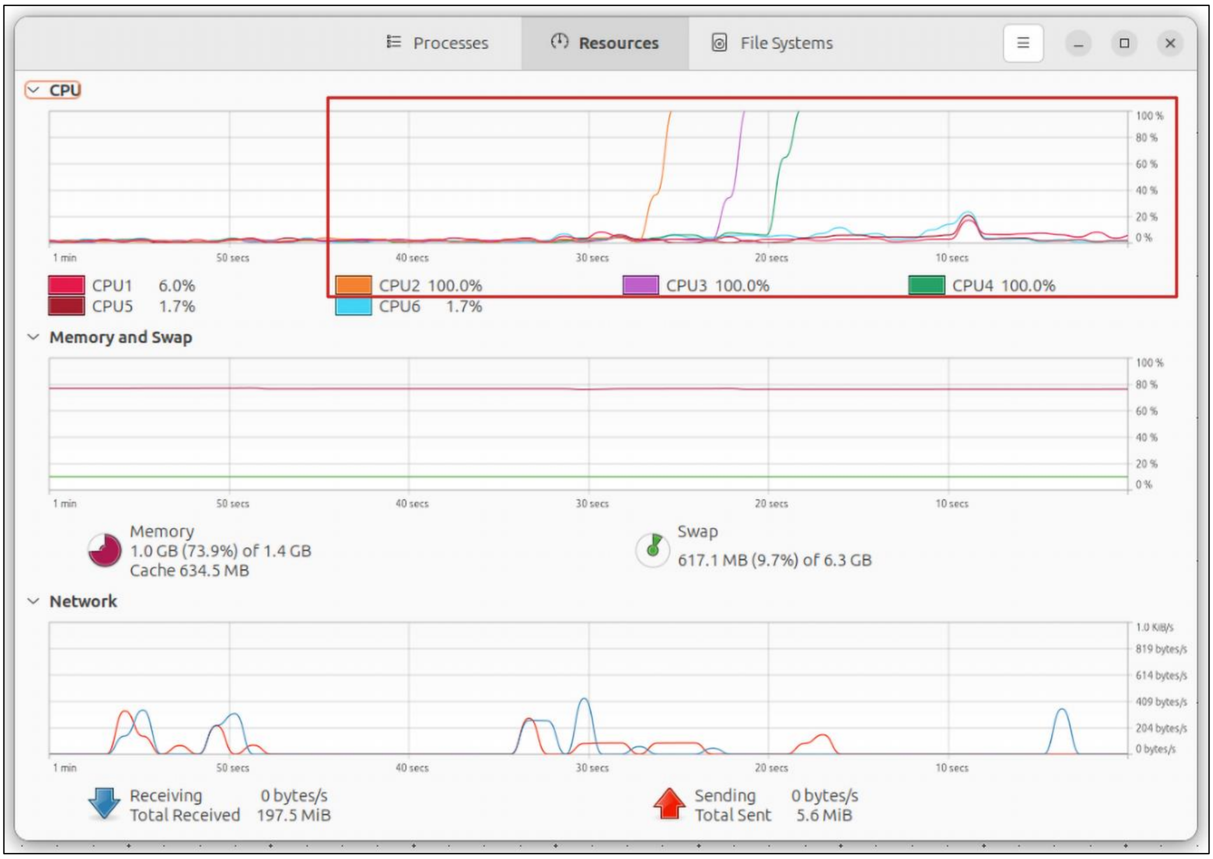
🟥 CPU1 2.9% 🟧 CPU2 100.0% 🟨 CPU3 96.2% 🟩 CPU4 100.0%
🟩 CPU5 2.0% 🟦 CPU6 3.9%

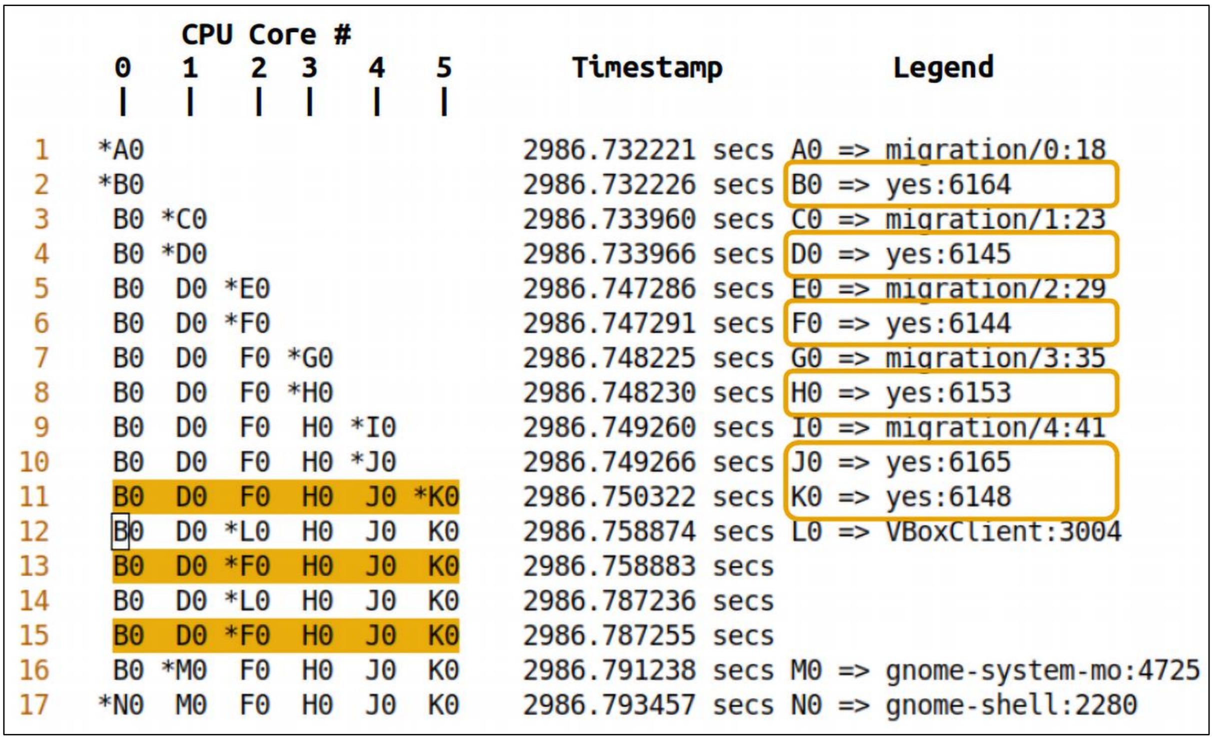
Memory
1.9 GB (46.3%) of 4.1 GB
Cache 2.2 GB

Swap
524.3 kB (0.0%) of 4.1 GB

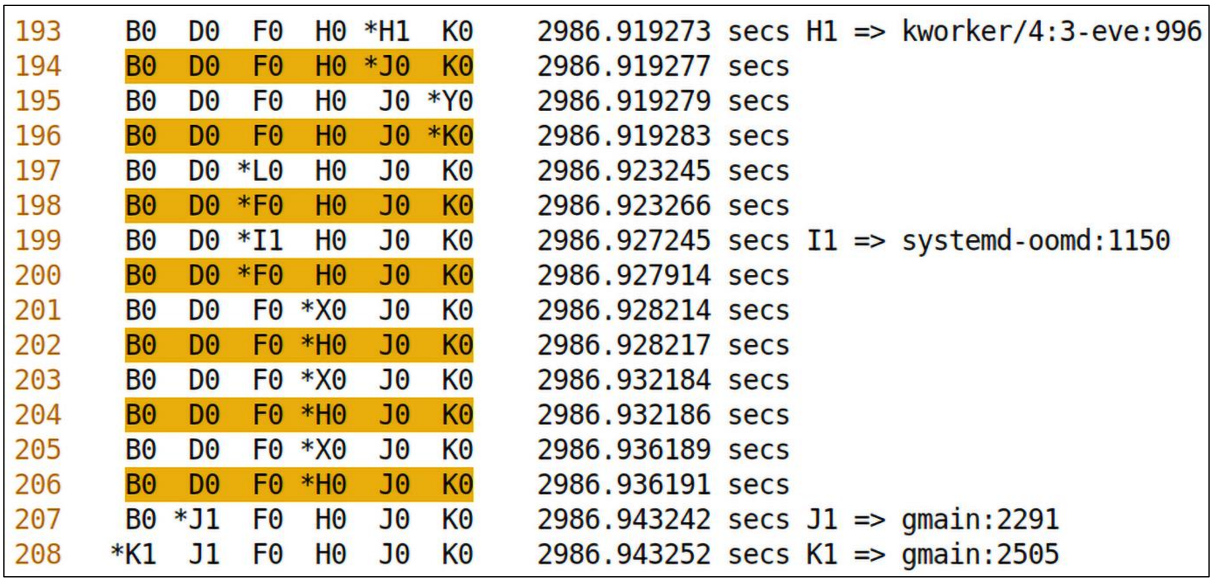
Receiving 953 bytes/s
Total Received 4.6 MiB

Sending 0 bytes/s
Total Sent 217.6 KiB

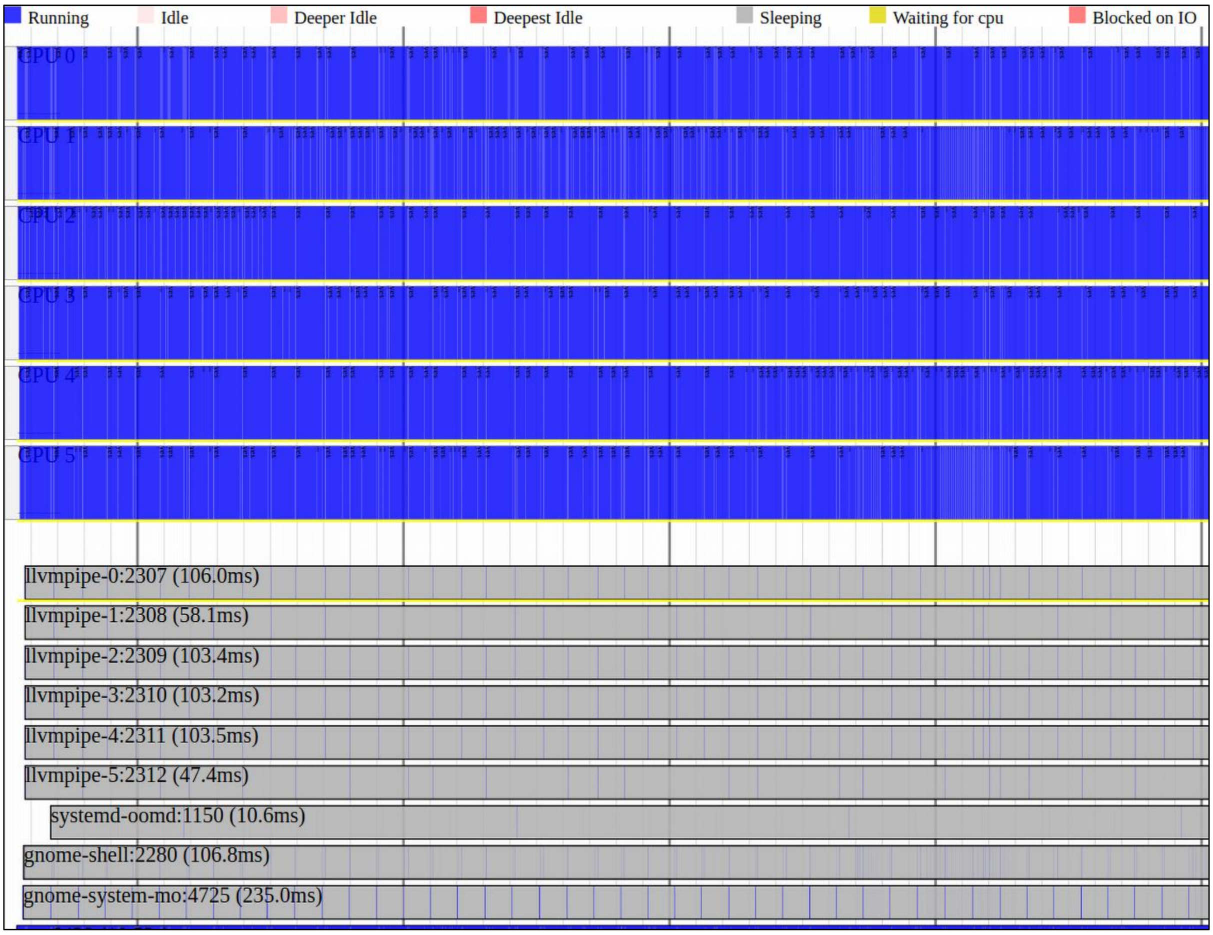


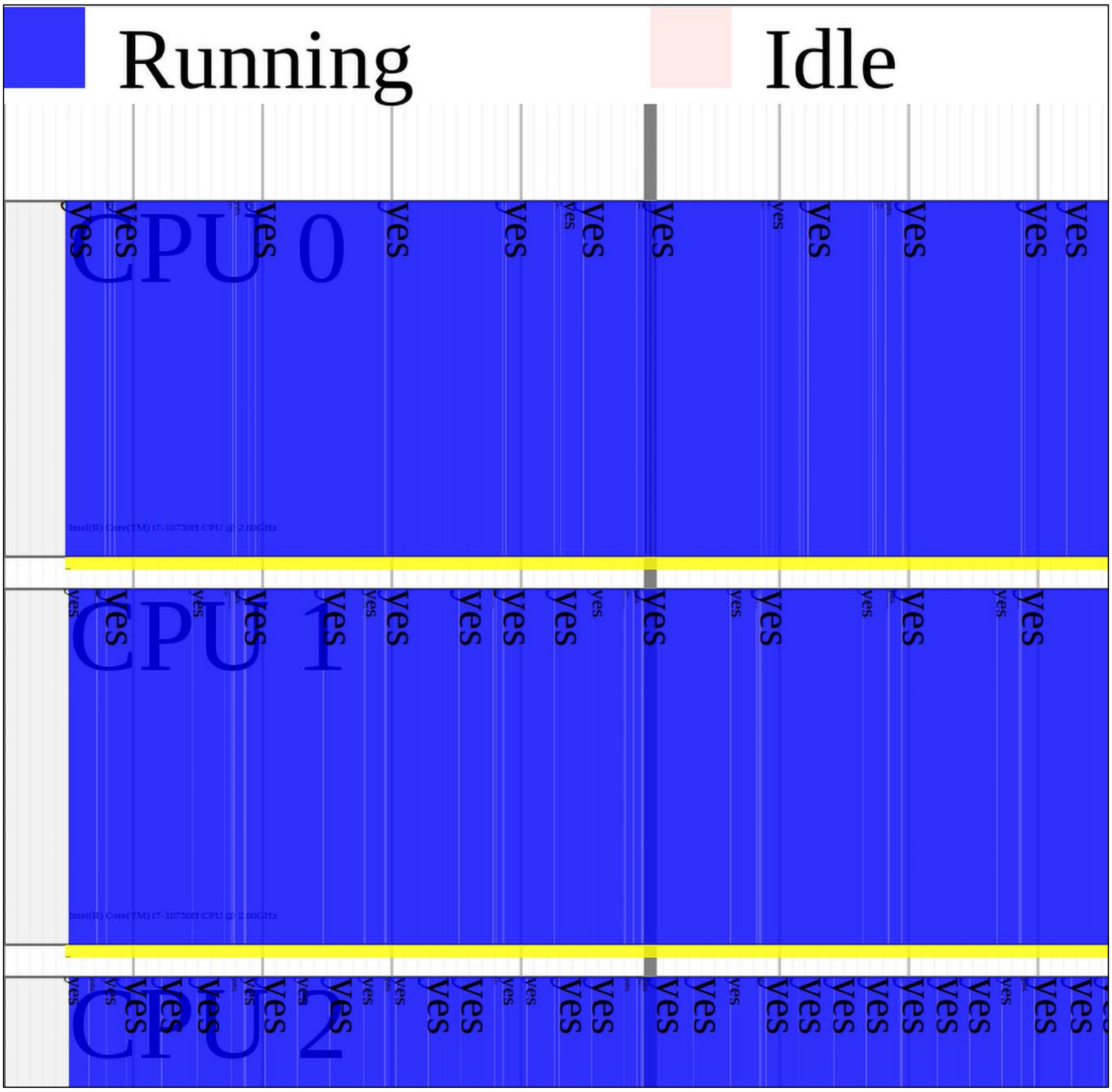


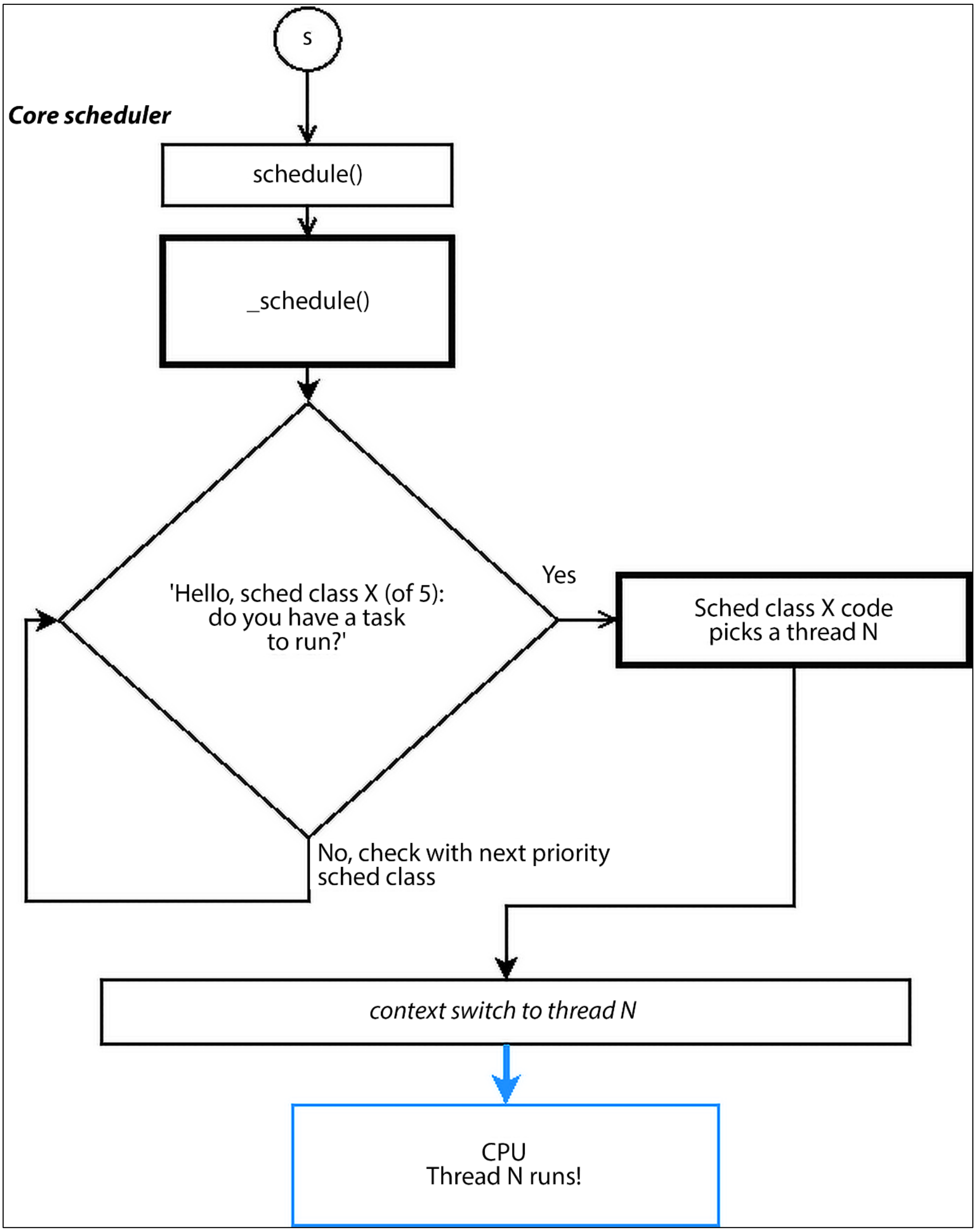
```
        CPU Core #
      0    1    2   3    4    5          Timestamp              Legend
      |    |    |   |    |    |

1    *A0                             2986.732221 secs A0 => migration/0:18
2    *B0                             2986.732226 secs B0 => yes:6164
3     B0  *C0                        2986.733960 secs C0 => migration/1:23
4     B0  *D0                        2986.733966 secs D0 => yes:6145
5     B0   D0  *E0                   2986.747286 secs E0 => migration/2:29
6     B0   D0  *F0                   2986.747291 secs F0 => yes:6144
7     B0   D0   F0  *G0              2986.748225 secs G0 => migration/3:35
8     B0   D0   F0  *H0              2986.748230 secs H0 => yes:6153
9     B0   D0   F0   H0  *I0         2986.749260 secs I0 => migration/4:41
10    B0   D0   F0   H0  *J0         2986.749266 secs J0 => yes:6165
11    B0   D0   F0   H0   J0  *K0    2986.750322 secs K0 => yes:6148
12    B0   D0  *L0   H0   J0   K0    2986.758874 secs L0 => VBoxClient:3004
13    B0   D0  *F0   H0   J0   K0    2986.758883 secs
14    B0   D0  *L0   H0   J0   K0    2986.787236 secs
15    B0   D0  *F0   H0   J0   K0    2986.787255 secs
16    B0  *M0   F0   H0   J0   K0    2986.791238 secs M0 => gnome-system-mo:4725
17   *N0   M0   F0   H0   J0   K0    2986.793457 secs N0 => gnome-shell:2280
```

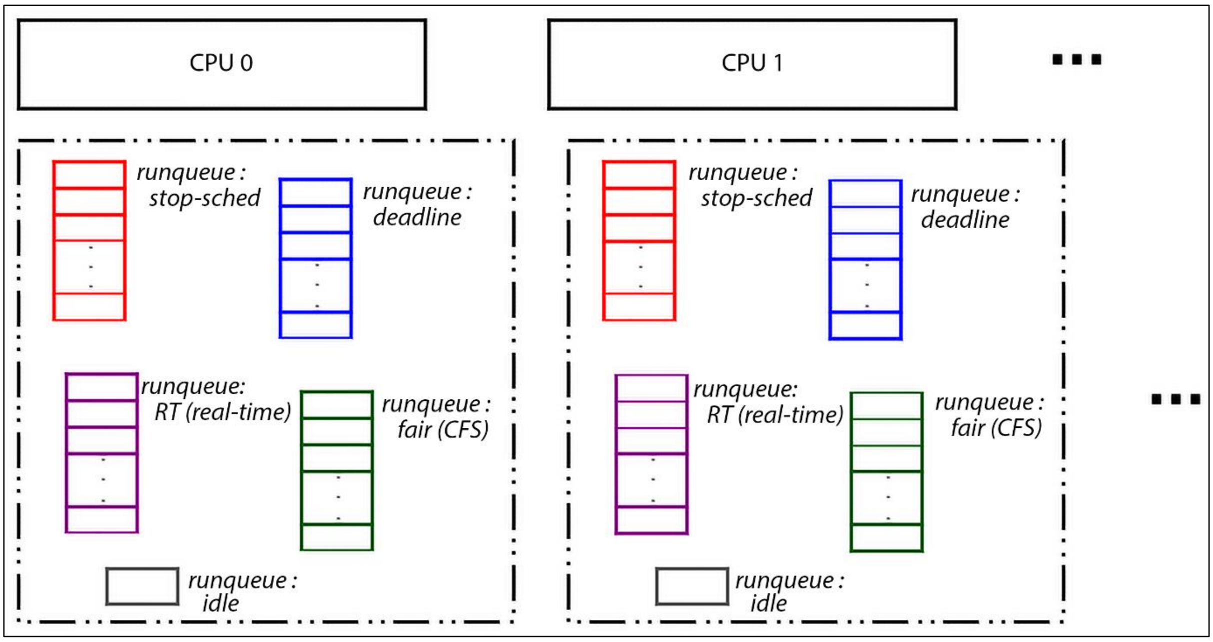


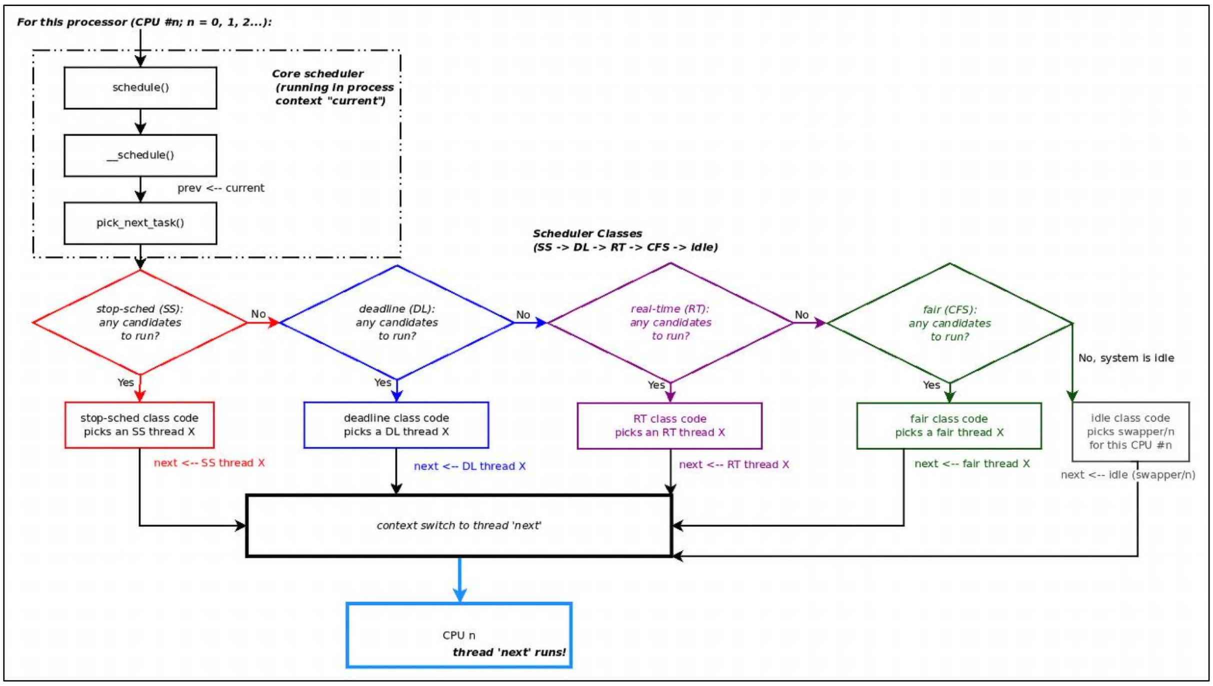
```
193     B0   D0   F0   H0  *H1   K0     2986.919273 secs H1 => kworker/4:3-eve:996
194     B0   D0   F0   H0  *J0   K0     2986.919277 secs
195     B0   D0   F0   H0   J0  *Y0     2986.919279 secs
196     B0   D0   F0   H0   J0  *K0     2986.919283 secs
197     B0   D0  *L0   H0   J0   K0     2986.923245 secs
198     B0   D0  *F0   H0   J0   K0     2986.923266 secs
199     B0   D0  *I1   H0   J0   K0     2986.927245 secs I1 => systemd-oomd:1150
200     B0   D0  *F0   H0   J0   K0     2986.927914 secs
201     B0   D0   F0  *X0   J0   K0     2986.928214 secs
202     B0   D0   F0  *H0   J0   K0     2986.928217 secs
203     B0   D0   F0  *X0   J0   K0     2986.932184 secs
204     B0   D0   F0  *H0   J0   K0     2986.932186 secs
205     B0   D0   F0  *X0   J0   K0     2986.936189 secs
206     B0   D0   F0  *H0   J0   K0     2986.936191 secs
207     B0  *J1   F0   H0   J0   K0     2986.943242 secs J1 => gmain:2291
208    *K1   J1   F0   H0   J0   K0     2986.943252 secs K1 => gmain:2505
```

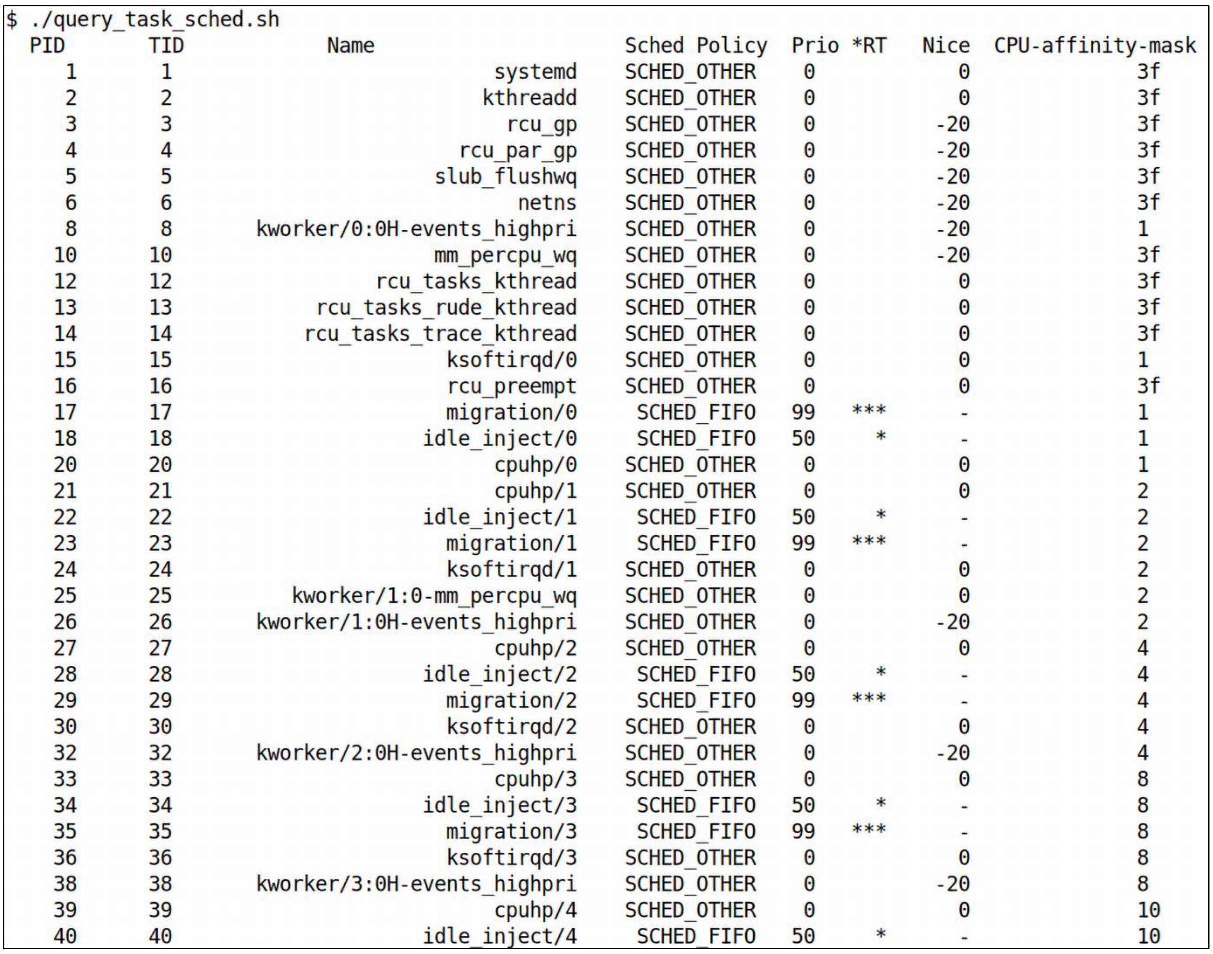



Running
Idle
CPU 0
yes
Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz
CPU 1
yes
Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz
CPU 2
yes

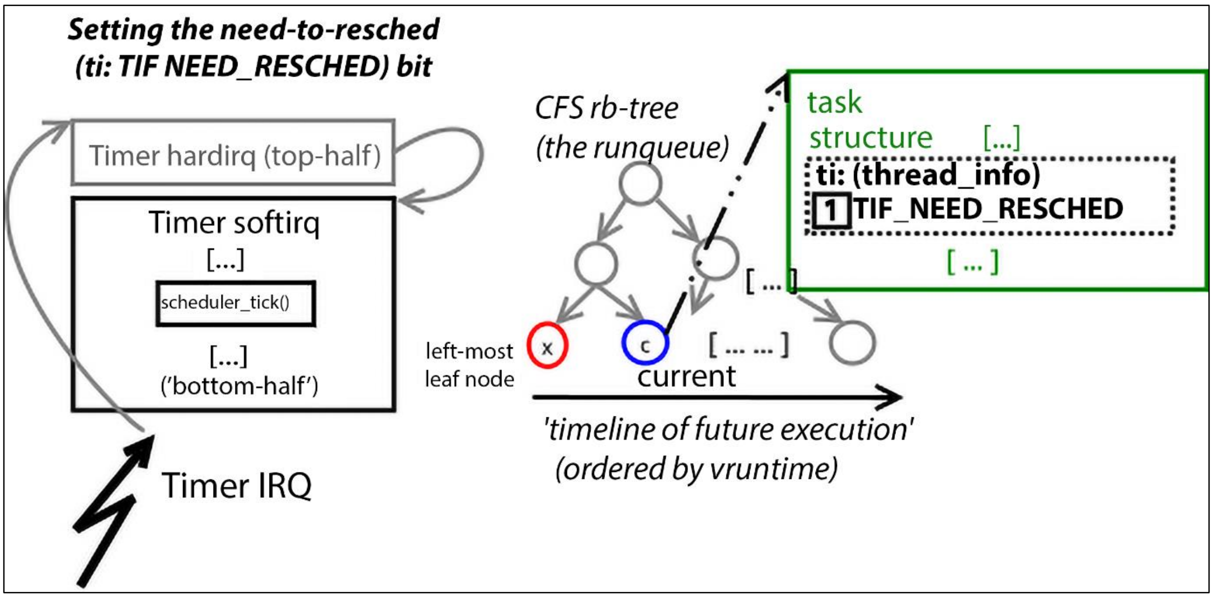

S
Core scheduler
schedule()
_schedule()
'Hello, sched class X (of 5):
do you have a task
to run?'
Yes
Sched class X code
picks a thread N
No, check with next priority
sched class
context switch to thread N
CPU
Thread N runs!


CPU 0
CPU 1
...
runqueue : stop-sched
runqueue : deadline
runqueue: RT (real-time)
runqueue : fair (CFS)
runqueue : idle
runqueue : stop-sched
runqueue : deadline
runqueue: RT (real-time)
runqueue : fair (CFS)
runqueue : idle
...

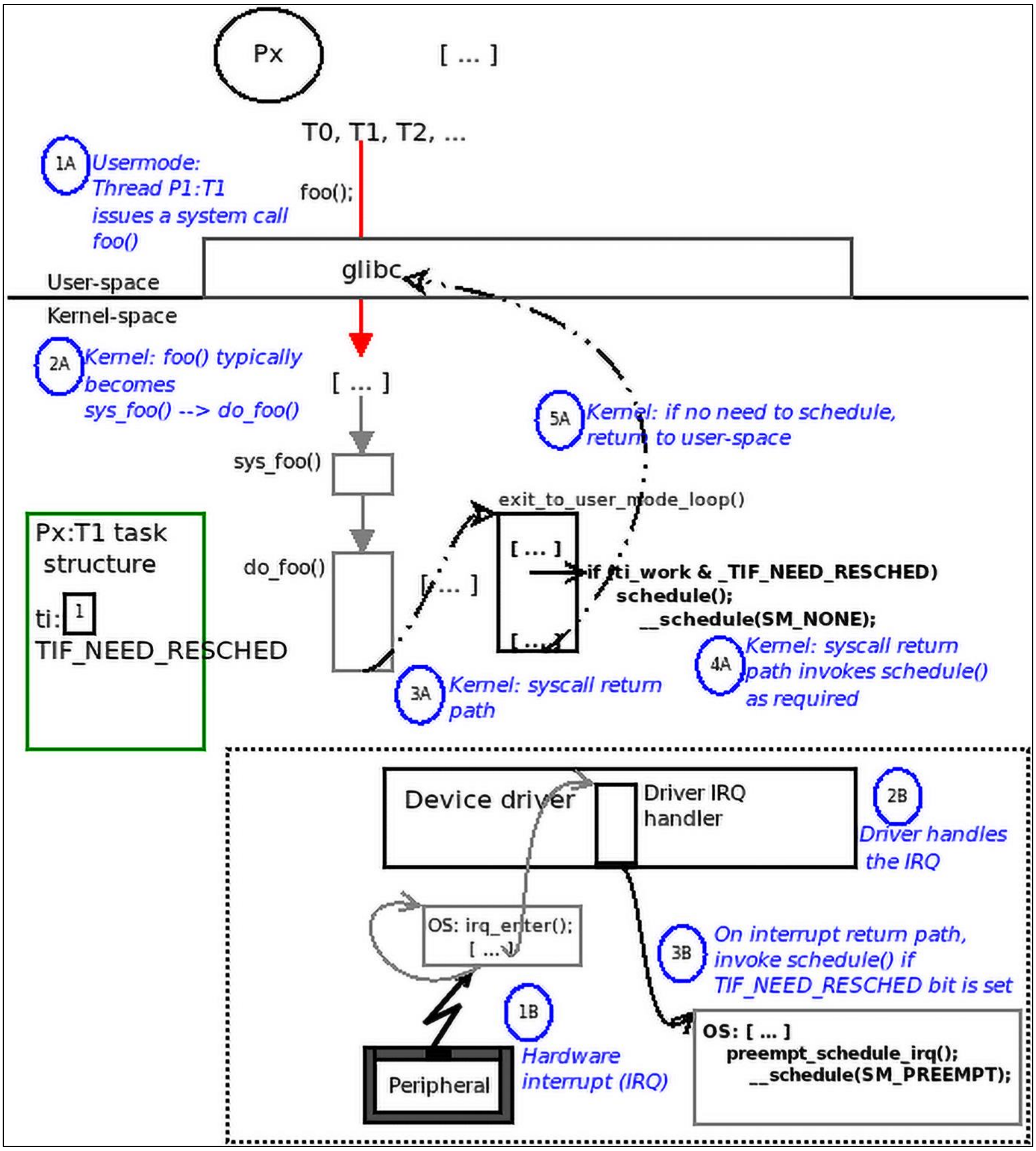

For this processor (CPU #n; n = 0, 1, 2...):
schedule()
Core scheduler (running in process context "current")
__schedule()
prev <-- current
pick_next_task()
Scheduler Classes (SS -> DL -> RT -> CFS -> idle)
stop-sched (SS): any candidates to run?
No
deadline (DL): any candidates to run?
No
real-time (RT): any candidates to run?
No
fair (CFS): any candidates to run?
No, system is idle
Yes
Yes
Yes
Yes
stop-sched class code picks an SS thread X
deadline class code picks a DL thread X
RT class code picks an RT thread X
fair class code picks a fair thread X
idle class code picks swapper/n for this CPU #n
next <-- SS thread X
next <-- DL thread X
next <-- RT thread X
next <-- fair thread X
next <-- idle (swapper/n)
context switch to thread 'next'
CPU n
thread 'next' runs!

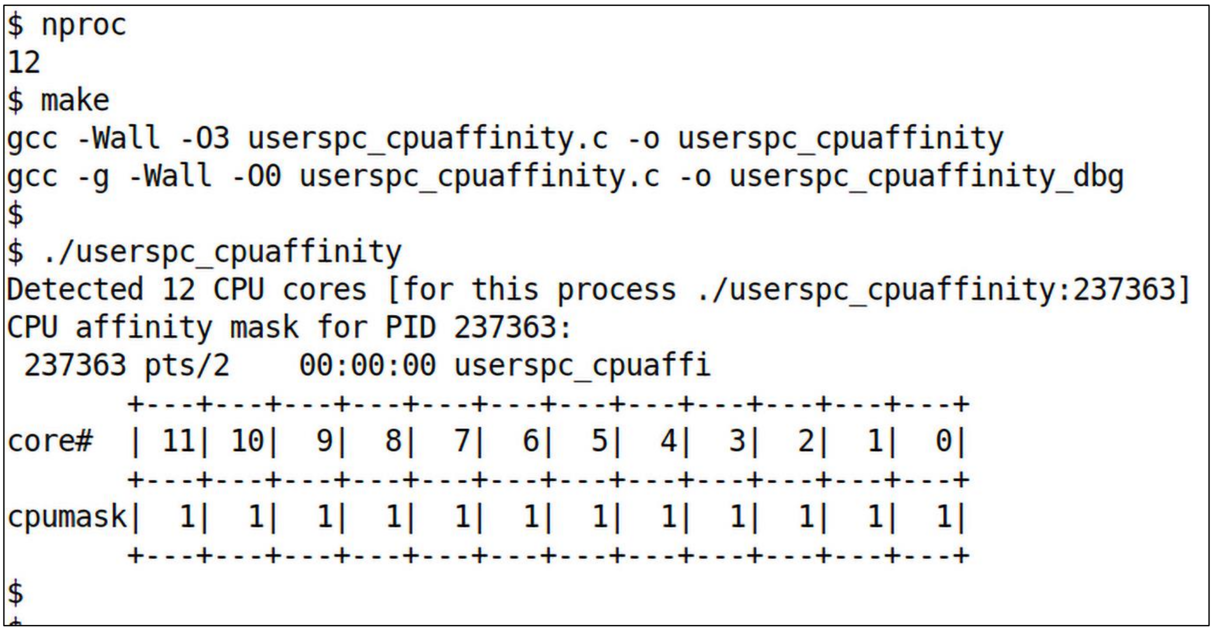


```
$ ./query_task_sched.sh
  PID       TID              Name                    Sched Policy  Prio *RT   Nice  CPU-affinity-mask
     1         1                         systemd     SCHED_OTHER   0           0             3f
     2         2                        kthreadd     SCHED_OTHER   0           0             3f
     3         3                          rcu_gp     SCHED_OTHER   0         -20             3f
     4         4                      rcu_par_gp     SCHED_OTHER   0         -20             3f
     5         5                    slub_flushwq     SCHED_OTHER   0         -20             3f
     6         6                           netns     SCHED_OTHER   0         -20             3f
     8         8      kworker/0:0H-events_highpri    SCHED_OTHER   0         -20             1
    10        10                     mm_percpu_wq    SCHED_OTHER   0         -20             3f
    12        12                rcu_tasks_kthread    SCHED_OTHER   0           0             3f
    13        13           rcu_tasks_rude_kthread    SCHED_OTHER   0           0             3f
    14        14          rcu_tasks_trace_kthread    SCHED_OTHER   0           0             3f
    15        15                      ksoftirqd/0    SCHED_OTHER   0           0             1
    16        16                      rcu_preempt    SCHED_OTHER   0           0             3f
    17        17                      migration/0     SCHED_FIFO  99   ***     -             1
    18        18                    idle_inject/0     SCHED_FIFO  50     *     -             1
    20        20                          cpuhp/0    SCHED_OTHER   0           0             1
    21        21                          cpuhp/1    SCHED_OTHER   0           0             2
    22        22                    idle_inject/1     SCHED_FIFO  50     *     -             2
    23        23                      migration/1     SCHED_FIFO  99   ***     -             2
    24        24                      ksoftirqd/1    SCHED_OTHER   0           0             2
    25        25          kworker/1:0-mm_percpu_wq   SCHED_OTHER   0           0             2
    26        26      kworker/1:0H-events_highpri    SCHED_OTHER   0         -20             2
    27        27                          cpuhp/2    SCHED_OTHER   0           0             4
    28        28                    idle_inject/2     SCHED_FIFO  50     *     -             4
    29        29                      migration/2     SCHED_FIFO  99   ***     -             4
    30        30                      ksoftirqd/2    SCHED_OTHER   0           0             4
    32        32      kworker/2:0H-events_highpri    SCHED_OTHER   0         -20             4
    33        33                          cpuhp/3    SCHED_OTHER   0           0             8
    34        34                    idle_inject/3     SCHED_FIFO  50     *     -             8
    35        35                      migration/3     SCHED_FIFO  99   ***     -             8
    36        36                      ksoftirqd/3    SCHED_OTHER   0           0             8
    38        38      kworker/3:0H-events_highpri    SCHED_OTHER   0         -20             8
    39        39                          cpuhp/4    SCHED_OTHER   0           0             10
    40        40                    idle_inject/4     SCHED_FIFO  50     *     -             10
```

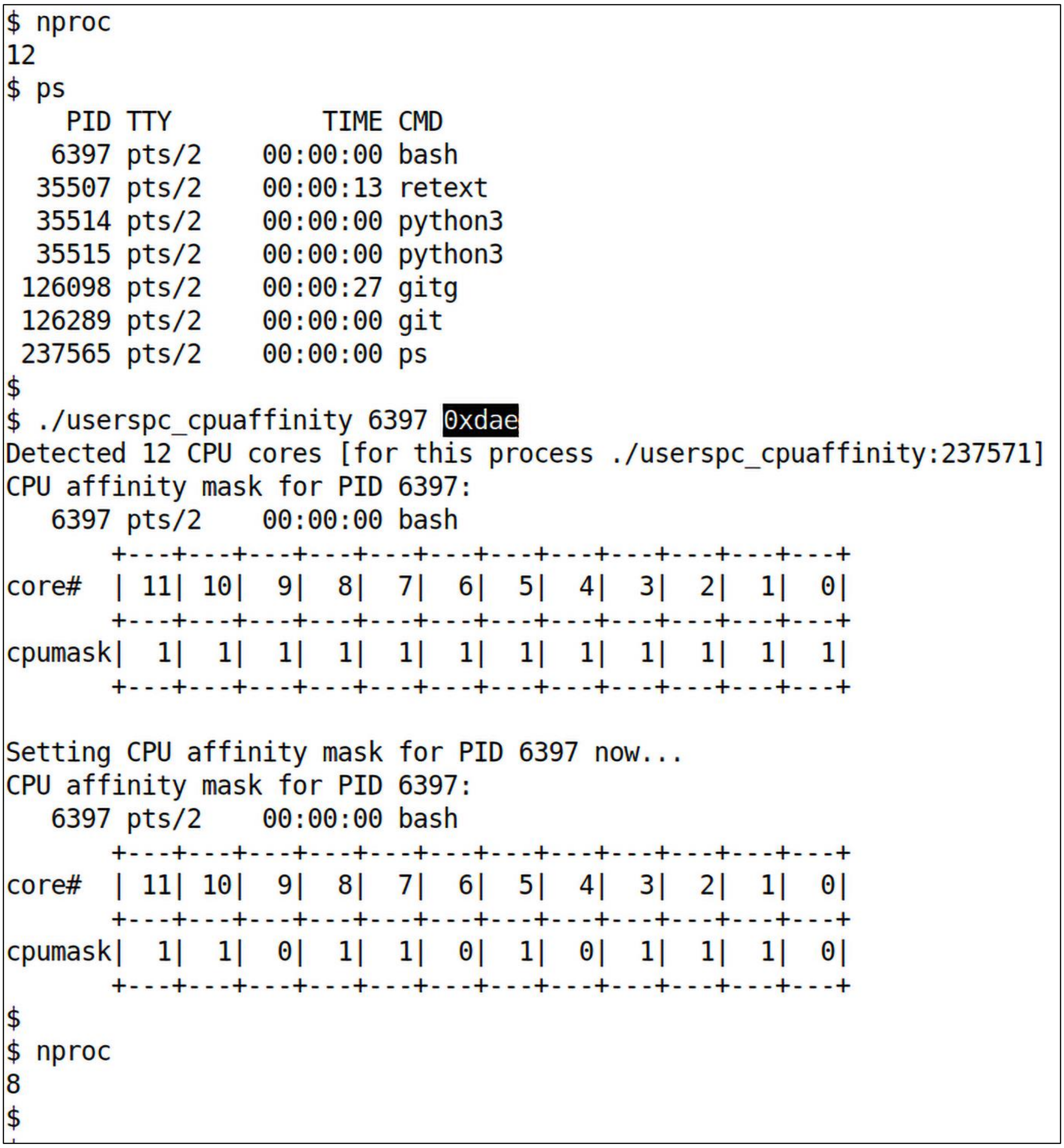


Setting the need-to-resched (ti: TIF NEED_RESCHED) bit

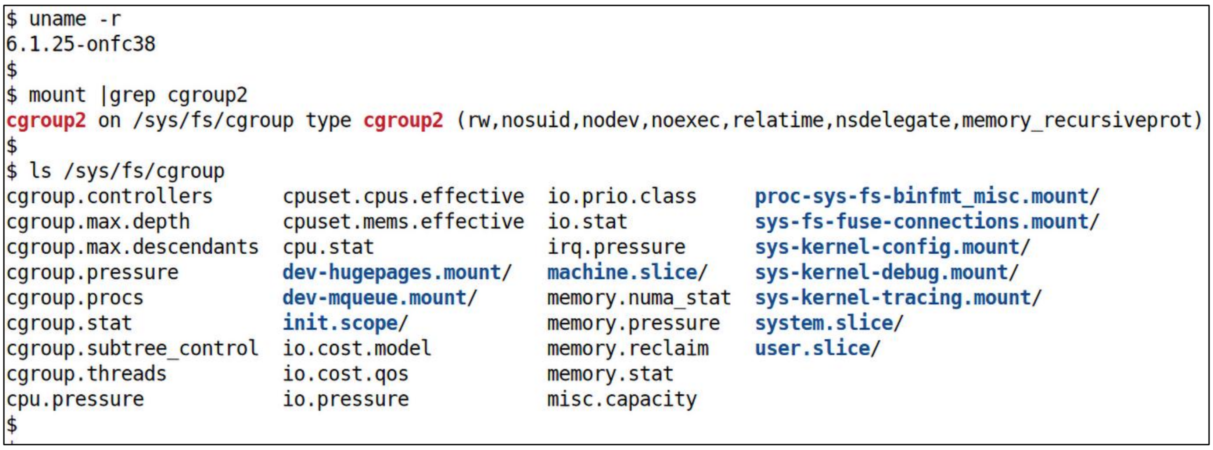

Px
[...]
T0, T1, T2, ...
1A
Usermode: Thread P1:T1 issues a system call foo()
foo();
glibc
User-space
Kernel-space
2A
Kernel: foo() typically becomes sys_foo() --> do_foo()
[...]
sys_foo()
do_foo()
Px:T1 task structure
ti: 1
TIF_NEED_RESCHED
5A
Kernel: if no need to schedule, return to user-space
exit_to_user_mode_loop()
[...]
[...]
if (ti_work & _TIF_NEED_RESCHED)
schedule();
__schedule(SM_NONE);
[...]
3A
Kernel: syscall return path
4A
Kernel: syscall return path invokes schedule() as required
Device driver
Driver IRQ handler
2B
Driver handles the IRQ
OS: irq_enter();
[...]
3B
On interrupt return path, invoke schedule() if TIF_NEED_RESCHED bit is set
1B
Hardware interrupt (IRQ)
Peripheral
OS: [...]
preempt_schedule_irq();
__schedule(SM_PREEMPT);



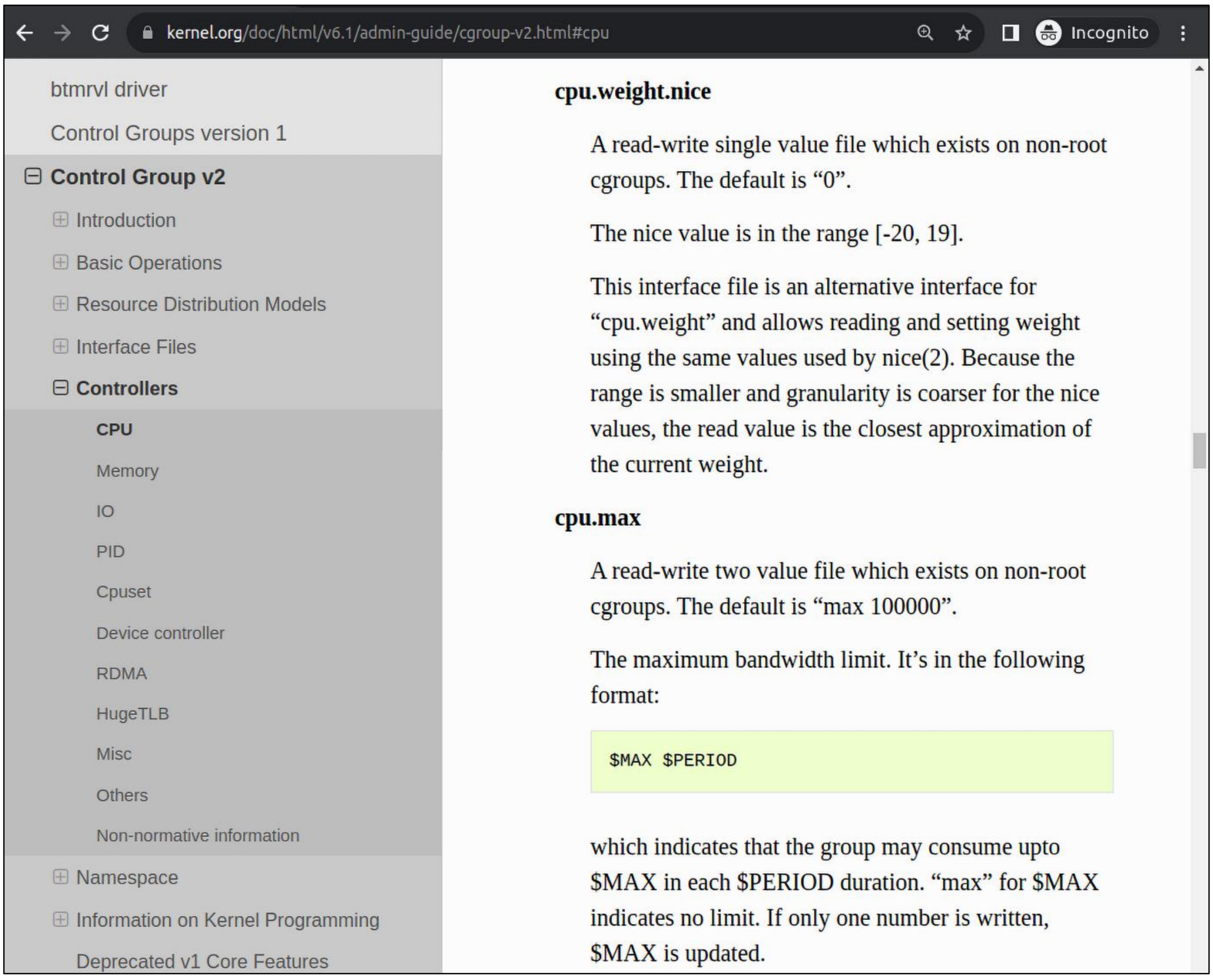
```
$ nproc
12
$ make
gcc -Wall -O3 userspc_cpuaffinity.c -o userspc_cpuaffinity
gcc -g -Wall -O0 userspc_cpuaffinity.c -o userspc_cpuaffinity_dbg
$
$ ./userspc_cpuaffinity
Detected 12 CPU cores [for this process ./userspc_cpuaffinity:237363]
CPU affinity mask for PID 237363:
 237363 pts/2    00:00:00 userspc_cpuaffi
       +---+---+---+---+---+---+---+---+---+---+---+---+
core#  | 11| 10|  9|  8|  7|  6|  5|  4|  3|  2|  1|  0|
       +---+---+---+---+---+---+---+---+---+---+---+---+
cpumask|  1|  1|  1|  1|  1|  1|  1|  1|  1|  1|  1|  1|
       +---+---+---+---+---+---+---+---+---+---+---+---+
$
```



```
$ nproc
12
$ ps
    PID TTY          TIME CMD
   6397 pts/2    00:00:00 bash
  35507 pts/2    00:00:13 retext
  35514 pts/2    00:00:00 python3
  35515 pts/2    00:00:00 python3
 126098 pts/2    00:00:27 gitg
 126289 pts/2    00:00:00 git
 237565 pts/2    00:00:00 ps
$
$ ./userspc_cpuaffinity 6397 0xdae
Detected 12 CPU cores [for this process ./userspc_cpuaffinity:237571]
CPU affinity mask for PID 6397:
   6397 pts/2    00:00:00 bash
       +---+---+---+---+---+---+---+---+---+---+---+---+
core#  | 11| 10|  9|  8|  7|  6|  5|  4|  3|  2|  1|  0|
       +---+---+---+---+---+---+---+---+---+---+---+---+
cpumask|  1|  1|  1|  1|  1|  1|  1|  1|  1|  1|  1|  1|
       +---+---+---+---+---+---+---+---+---+---+---+---+

Setting CPU affinity mask for PID 6397 now...
CPU affinity mask for PID 6397:
   6397 pts/2    00:00:00 bash
       +---+---+---+---+---+---+---+---+---+---+---+---+
core#  | 11| 10|  9|  8|  7|  6|  5|  4|  3|  2|  1|  0|
       +---+---+---+---+---+---+---+---+---+---+---+---+
cpumask|  1|  1|  0|  1|  1|  0|  1|  0|  1|  1|  1|  0|
       +---+---+---+---+---+---+---+---+---+---+---+---+
$
$ nproc
8
$
```



```
$ uname -r
6.1.25-onfc38
$
$ mount |grep cgroup2
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot)
$
$ ls /sys/fs/cgroup
cgroup.controllers      cpuset.cpus.effective  io.prio.class      proc-sys-fs-binfmt_misc.mount/
cgroup.max.depth        cpuset.mems.effective  io.stat            sys-fs-fuse-connections.mount/
cgroup.max.descendants  cpu.stat               irq.pressure       sys-kernel-config.mount/
cgroup.pressure         dev-hugepages.mount/   machine.slice/     sys-kernel-debug.mount/
cgroup.procs            dev-mqueue.mount/      memory.numa_stat   sys-kernel-tracing.mount/
cgroup.stat             init.scope/            memory.pressure    system.slice/
cgroup.subtree_control  io.cost.model          memory.reclaim     user.slice/
cgroup.threads          io.cost.qos            memory.stat
cpu.pressure            io.pressure            misc.capacity
$
```



cpu.weight.nice

A read-write single value file which exists on non-root cgroups. The default is “0”.

The nice value is in the range [-20, 19].

This interface file is an alternative interface for “cpu.weight” and allows reading and setting weight using the same values used by nice(2). Because the range is smaller and granularity is coarser for the nice values, the read value is the closest approximation of the current weight.

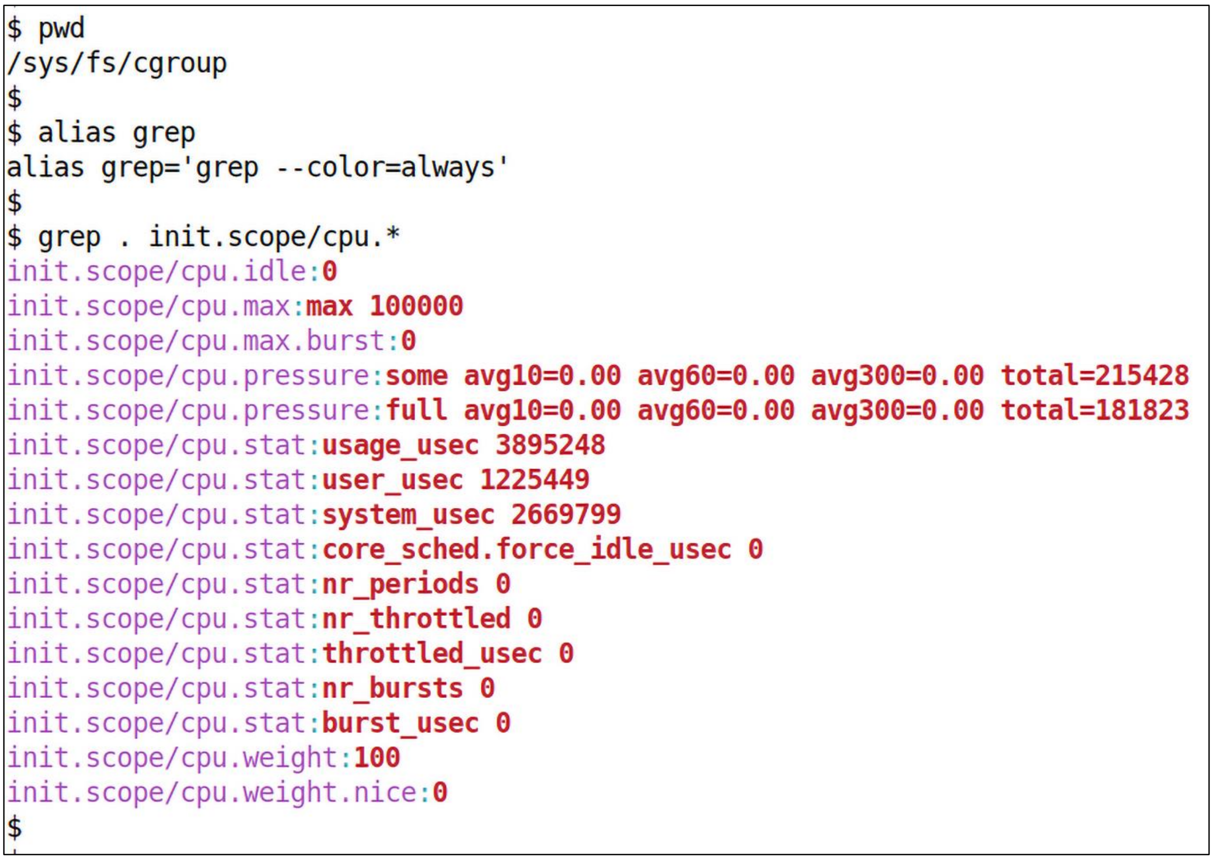
cpu.max

A read-write two value file which exists on non-root cgroups. The default is “max 100000”.

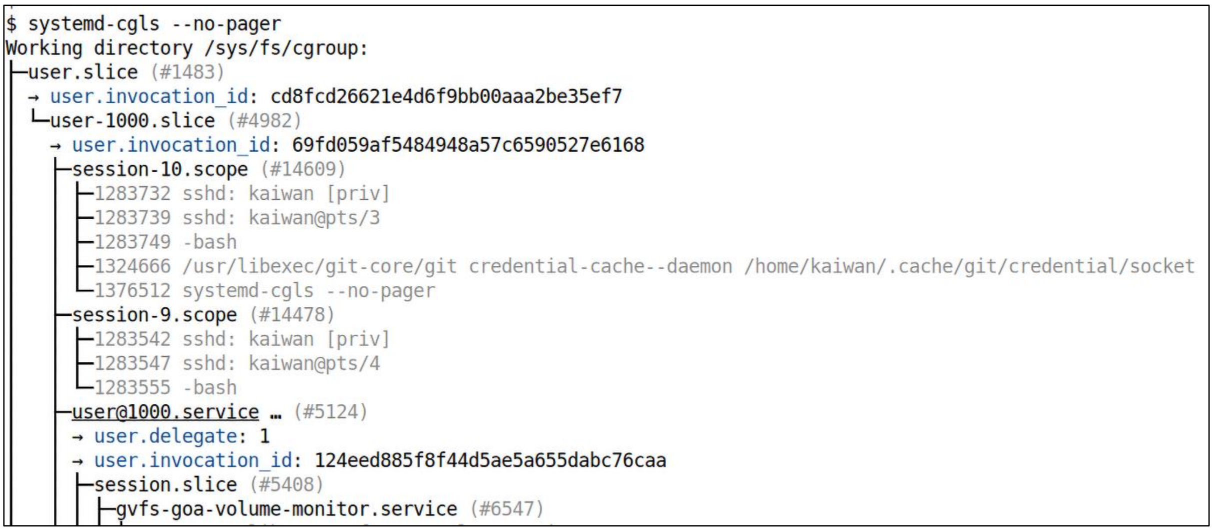
The maximum bandwidth limit. It’s in the following format:

```
$MAX $PERIOD
```

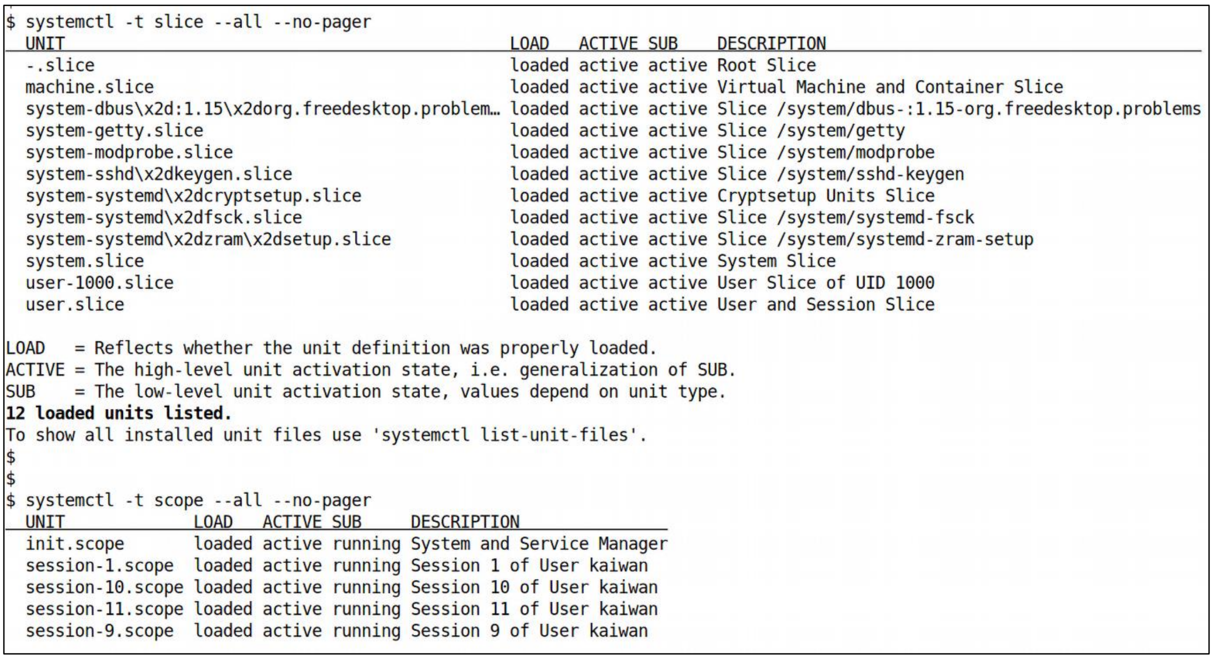
which indicates that the group may consume upto $MAX in each $PERIOD duration. “max” for $MAX indicates no limit. If only one number is written, $MAX is updated.



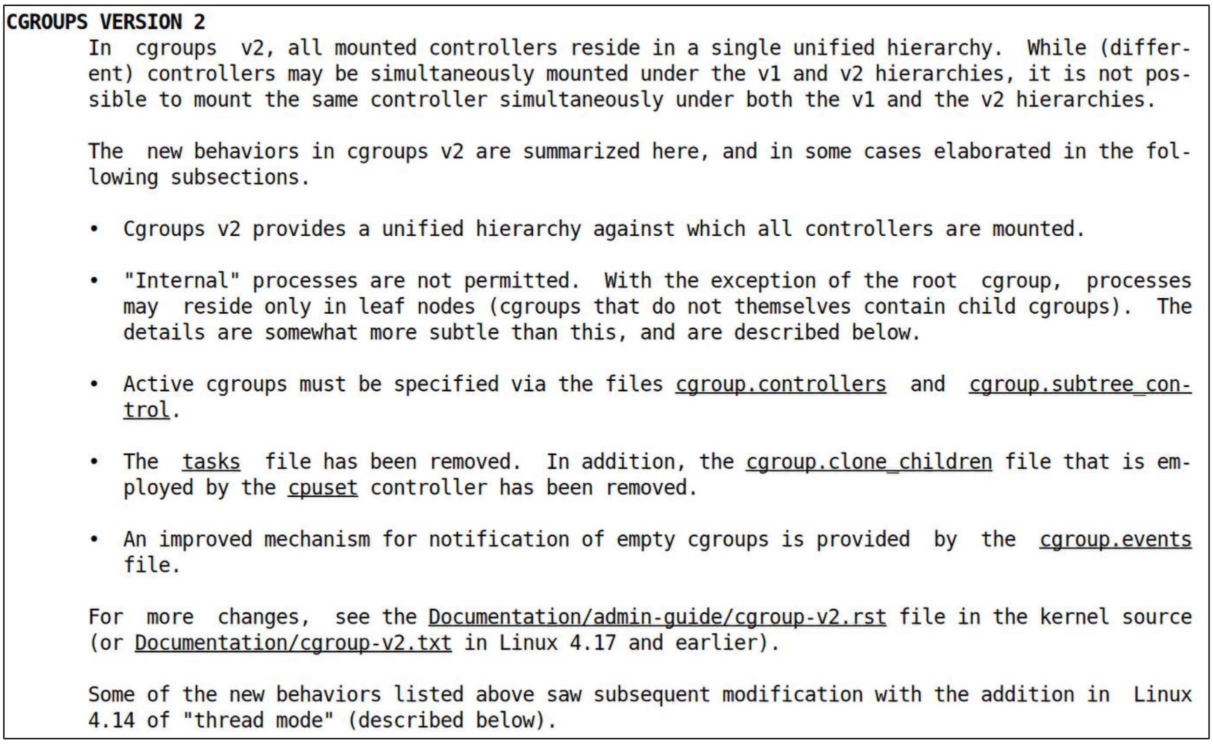
```
$ pwd
/sys/fs/cgroup
$
$ alias grep
alias grep='grep --color=always'
$
$ grep . init.scope/cpu.*
init.scope/cpu.idle:0
init.scope/cpu.max:max 100000
init.scope/cpu.max.burst:0
init.scope/cpu.pressure:some avg10=0.00 avg60=0.00 avg300=0.00 total=215428
init.scope/cpu.pressure:full avg10=0.00 avg60=0.00 avg300=0.00 total=181823
init.scope/cpu.stat:usage_usec 3895248
init.scope/cpu.stat:user_usec 1225449
init.scope/cpu.stat:system_usec 2669799
init.scope/cpu.stat:core_sched.force_idle_usec 0
init.scope/cpu.stat:nr_periods 0
init.scope/cpu.stat:nr_throttled 0
init.scope/cpu.stat:throttled_usec 0
init.scope/cpu.stat:nr_bursts 0
init.scope/cpu.stat:burst_usec 0
init.scope/cpu.weight:100
init.scope/cpu.weight.nice:0
$
```



```
$ systemd-cgls --no-pager
Working directory /sys/fs/cgroup:
├─user.slice (#1483)
│ → user.invocation_id: cd8fcd26621e4d6f9bb00aaa2be35ef7
│ └─user-1000.slice (#4982)
│   → user.invocation_id: 69fd059af5484948a57c6590527e6168
│   ├─session-10.scope (#14609)
│   │ ├─1283732 sshd: kaiwan [priv]
│   │ ├─1283739 sshd: kaiwan@pts/3
│   │ ├─1283749 -bash
│   │ ├─1324666 /usr/libexec/git-core/git credential-cache--daemon /home/kaiwan/.cache/git/credential/socket
│   │ └─1376512 systemd-cgls --no-pager
│   ├─session-9.scope (#14478)
│   │ ├─1283542 sshd: kaiwan [priv]
│   │ ├─1283547 sshd: kaiwan@pts/4
│   │ └─1283555 -bash
│   ├─user@1000.service … (#5124)
│   │ → user.delegate: 1
│   │ → user.invocation_id: 124eed885f8f44d5ae5a655dabc76caa
│   │ ├─session.slice (#5408)
│   │ │ ├─gvfs-goa-volume-monitor.service (#6547)
```



```
$ systemctl -t slice --all --no-pager
  UNIT                                               LOAD   ACTIVE SUB    DESCRIPTION
  -.slice                                            loaded active active Root Slice
  machine.slice                                      loaded active active Virtual Machine and Container Slice
  system-dbus\x2d:1.15\x2dorg.freedesktop.problem… loaded active active Slice /system/dbus-:1.15-org.freedesktop.problems
  system-getty.slice                                 loaded active active Slice /system/getty
  system-modprobe.slice                              loaded active active Slice /system/modprobe
  system-sshd\x2dkeygen.slice                        loaded active active Slice /system/sshd-keygen
  system-systemd\x2dcryptsetup.slice                 loaded active active Cryptsetup Units Slice
  system-systemd\x2dfsck.slice                       loaded active active Slice /system/systemd-fsck
  system-systemd\x2dzram\x2dsetup.slice              loaded active active Slice /system/systemd-zram-setup
  system.slice                                       loaded active active System Slice
  user-1000.slice                                    loaded active active User Slice of UID 1000
  user.slice                                         loaded active active User and Session Slice

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.
12 loaded units listed.
To show all installed unit files use 'systemctl list-unit-files'.
$
$
$ systemctl -t scope --all --no-pager
  UNIT             LOAD   ACTIVE SUB     DESCRIPTION
  init.scope       loaded active running System and Service Manager
  session-1.scope  loaded active running Session 1 of User kaiwan
  session-10.scope loaded active running Session 10 of User kaiwan
  session-11.scope loaded active running Session 11 of User kaiwan
  session-9.scope  loaded active running Session 9 of User kaiwan
```



CGROUPS VERSION 2

In cgroups v2, all mounted controllers reside in a single unified hierarchy. While (different) controllers may be simultaneously mounted under the v1 and v2 hierarchies, it is not possible to mount the same controller simultaneously under both the v1 and the v2 hierarchies.

The new behaviors in cgroups v2 are summarized here, and in some cases elaborated in the following subsections.

- Cgroups v2 provides a unified hierarchy against which all controllers are mounted.

- "Internal" processes are not permitted. With the exception of the root cgroup, processes may reside only in leaf nodes (cgroups that do not themselves contain child cgroups). The details are somewhat more subtle than this, and are described below.

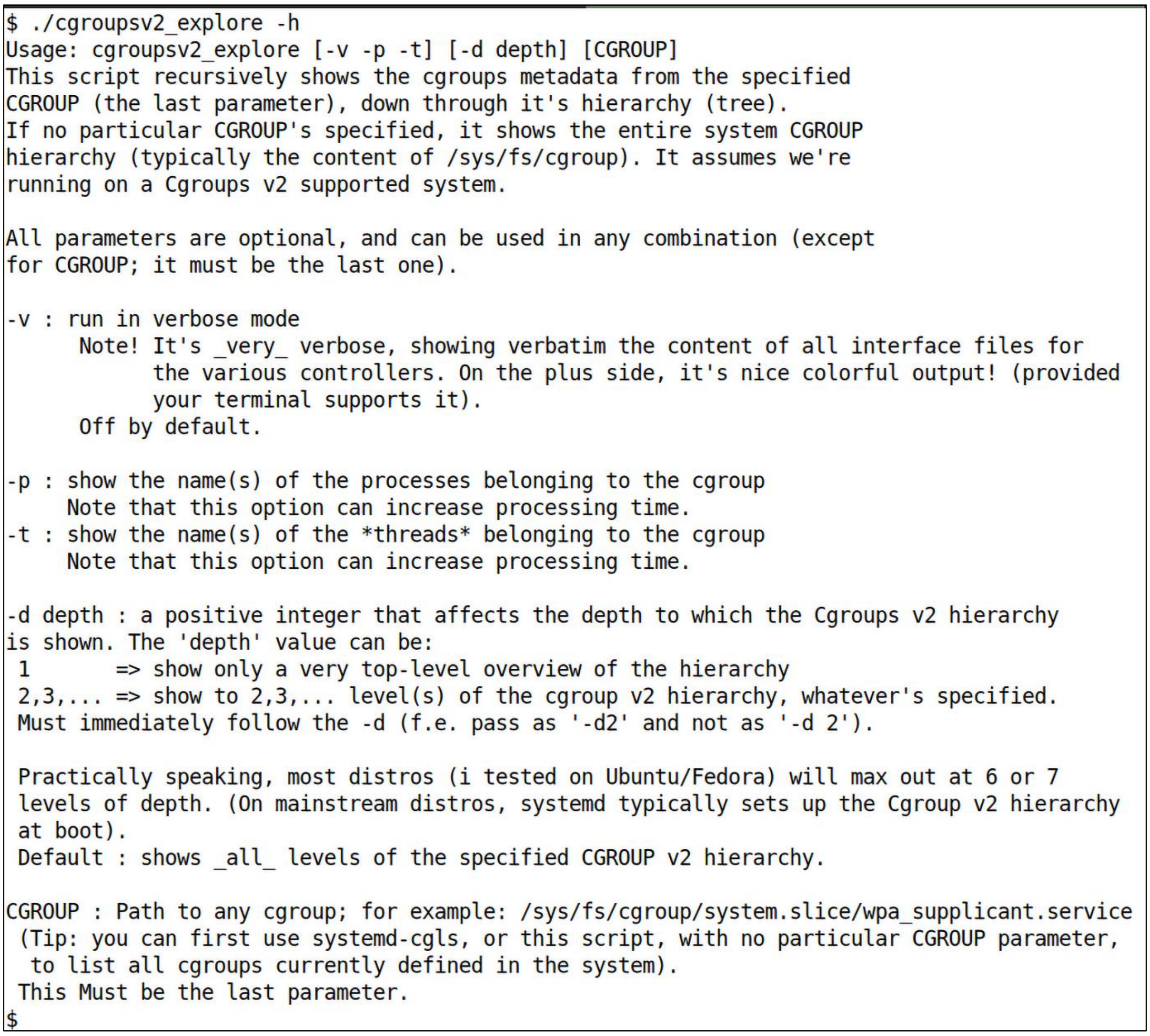
- Active cgroups must be specified via the files cgroup.controllers and cgroup.subtree_control.

- The tasks file has been removed. In addition, the cgroup.clone_children file that is employed by the cpuset controller has been removed.

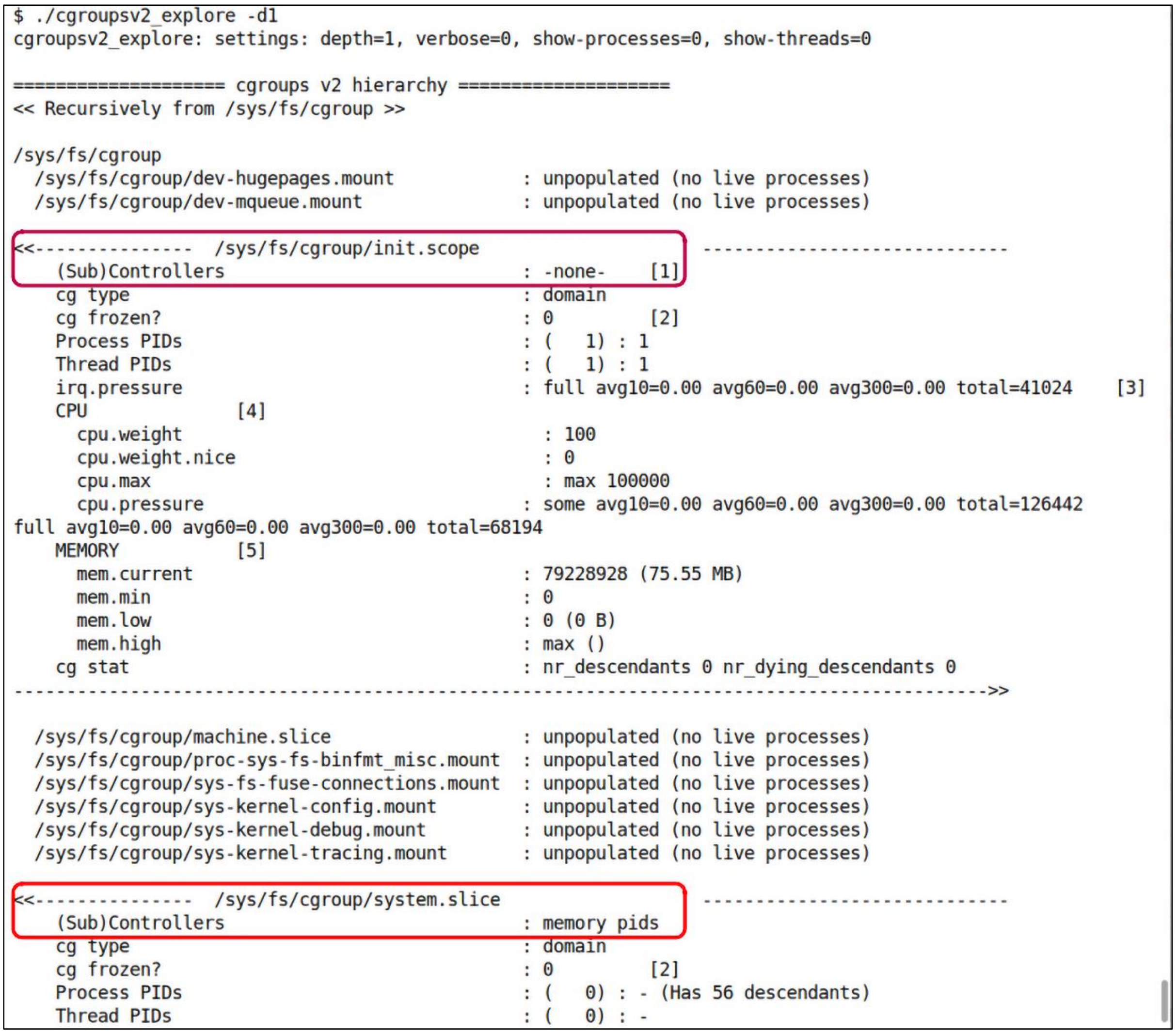
- An improved mechanism for notification of empty cgroups is provided by the cgroup.events file.

For more changes, see the Documentation/admin-guide/cgroup-v2.rst file in the kernel source (or Documentation/cgroup-v2.txt in Linux 4.17 and earlier).

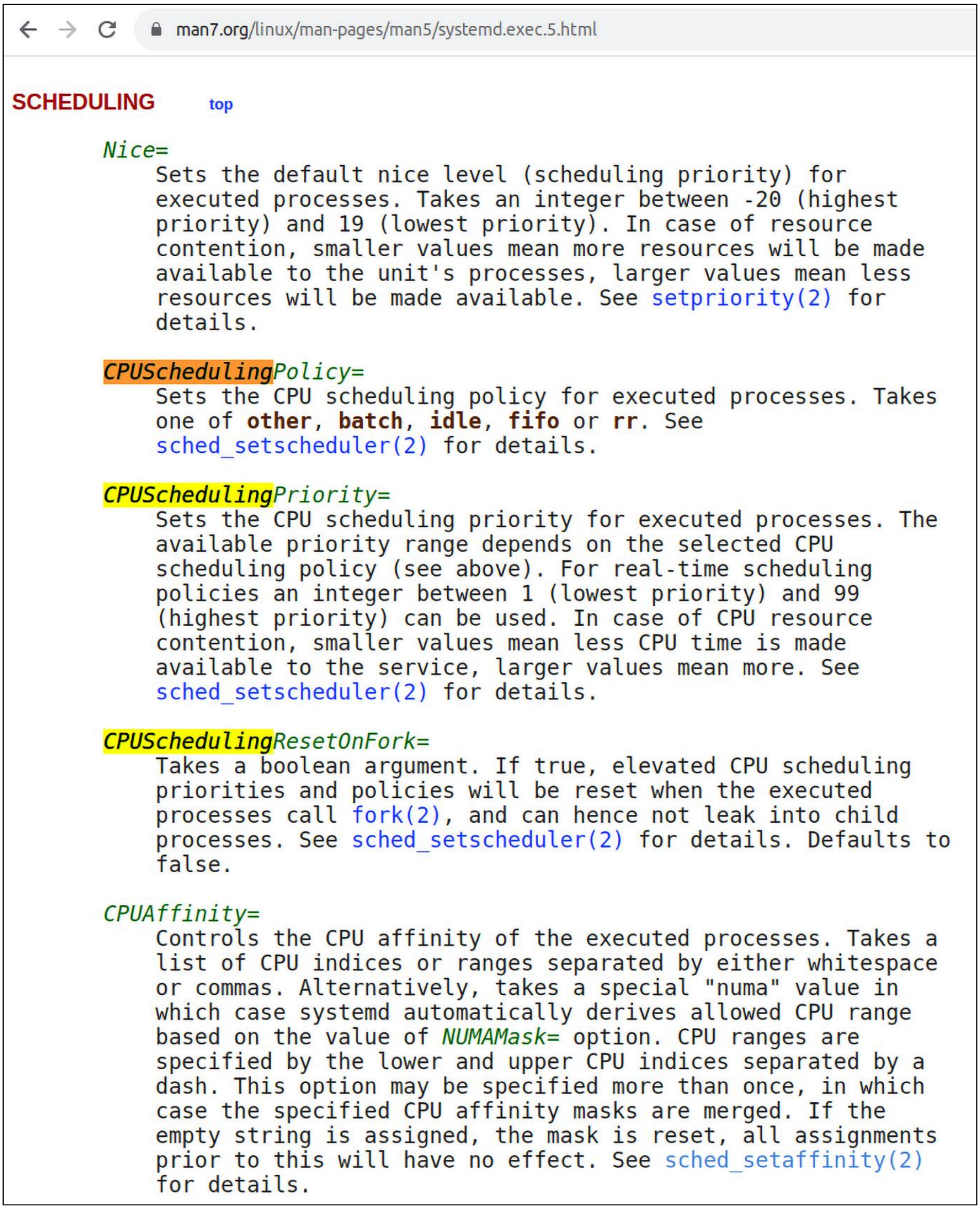
Some of the new behaviors listed above saw subsequent modification with the addition in Linux 4.14 of "thread mode" (described below).



```
$ ./cgroupsv2_explore -h
Usage: cgroupsv2_explore [-v -p -t] [-d depth] [CGROUP]
This script recursively shows the cgroups metadata from the specified
CGROUP (the last parameter), down through it's hierarchy (tree).
If no particular CGROUP's specified, it shows the entire system CGROUP
hierarchy (typically the content of /sys/fs/cgroup). It assumes we're
running on a Cgroups v2 supported system.

All parameters are optional, and can be used in any combination (except
for CGROUP; it must be the last one).

-v : run in verbose mode
      Note! It's _very_ verbose, showing verbatim the content of all interface files for
            the various controllers. On the plus side, it's nice colorful output! (provided
            your terminal supports it).
      Off by default.

-p : show the name(s) of the processes belonging to the cgroup
     Note that this option can increase processing time.
-t : show the name(s) of the *threads* belonging to the cgroup
     Note that this option can increase processing time.

-d depth : a positive integer that affects the depth to which the Cgroups v2 hierarchy
is shown. The 'depth' value can be:
 1       => show only a very top-level overview of the hierarchy
 2,3,... => show to 2,3,... level(s) of the cgroup v2 hierarchy, whatever's specified.
 Must immediately follow the -d (f.e. pass as '-d2' and not as '-d 2').

 Practically speaking, most distros (i tested on Ubuntu/Fedora) will max out at 6 or 7
 levels of depth. (On mainstream distros, systemd typically sets up the Cgroup v2 hierarchy
 at boot).
 Default : shows _all_ levels of the specified CGROUP v2 hierarchy.

CGROUP : Path to any cgroup; for example: /sys/fs/cgroup/system.slice/wpa_supplicant.service
 (Tip: you can first use systemd-cgls, or this script, with no particular CGROUP parameter,
  to list all cgroups currently defined in the system).
 This Must be the last parameter.
$
```



```
$ ./cgroupsv2_explore -d1
cgroupsv2_explore: settings: depth=1, verbose=0, show-processes=0, show-threads=0

==================== cgroups v2 hierarchy ====================
<< Recursively from /sys/fs/cgroup >>

/sys/fs/cgroup
  /sys/fs/cgroup/dev-hugepages.mount            : unpopulated (no live processes)
  /sys/fs/cgroup/dev-mqueue.mount               : unpopulated (no live processes)

<<---------------  /sys/fs/cgroup/init.scope                             ----------------------------
    (Sub)Controllers                            : -none-     [1]
    cg type                                     : domain
    cg frozen?                                  : 0          [2]
    Process PIDs                                : (    1) : 1
    Thread PIDs                                 : (    1) : 1
    irq.pressure                                : full avg10=0.00 avg60=0.00 avg300=0.00 total=41024    [3]
    CPU               [4]
      cpu.weight                                  : 100
      cpu.weight.nice                             : 0
      cpu.max                                     : max 100000
      cpu.pressure                              : some avg10=0.00 avg60=0.00 avg300=0.00 total=126442
full avg10=0.00 avg60=0.00 avg300=0.00 total=68194
    MEMORY            [5]
      mem.current                               : 79228928 (75.55 MB)
      mem.min                                   : 0
      mem.low                                   : 0 (0 B)
      mem.high                                  : max ()
    cg stat                                     : nr_descendants 0 nr_dying_descendants 0
----------------------------------------------------------------------------------------------->>

  /sys/fs/cgroup/machine.slice                  : unpopulated (no live processes)
  /sys/fs/cgroup/proc-sys-fs-binfmt_misc.mount  : unpopulated (no live processes)
  /sys/fs/cgroup/sys-fs-fuse-connections.mount  : unpopulated (no live processes)
  /sys/fs/cgroup/sys-kernel-config.mount        : unpopulated (no live processes)
  /sys/fs/cgroup/sys-kernel-debug.mount         : unpopulated (no live processes)
  /sys/fs/cgroup/sys-kernel-tracing.mount       : unpopulated (no live processes)

<<---------------  /sys/fs/cgroup/system.slice                           ----------------------------
    (Sub)Controllers                            : memory pids
    cg type                                     : domain
    cg frozen?                                  : 0          [2]
    Process PIDs                                : (    0) : - (Has 56 descendants)
    Thread PIDs                                 : (    0) : -
```



SCHEDULING

top

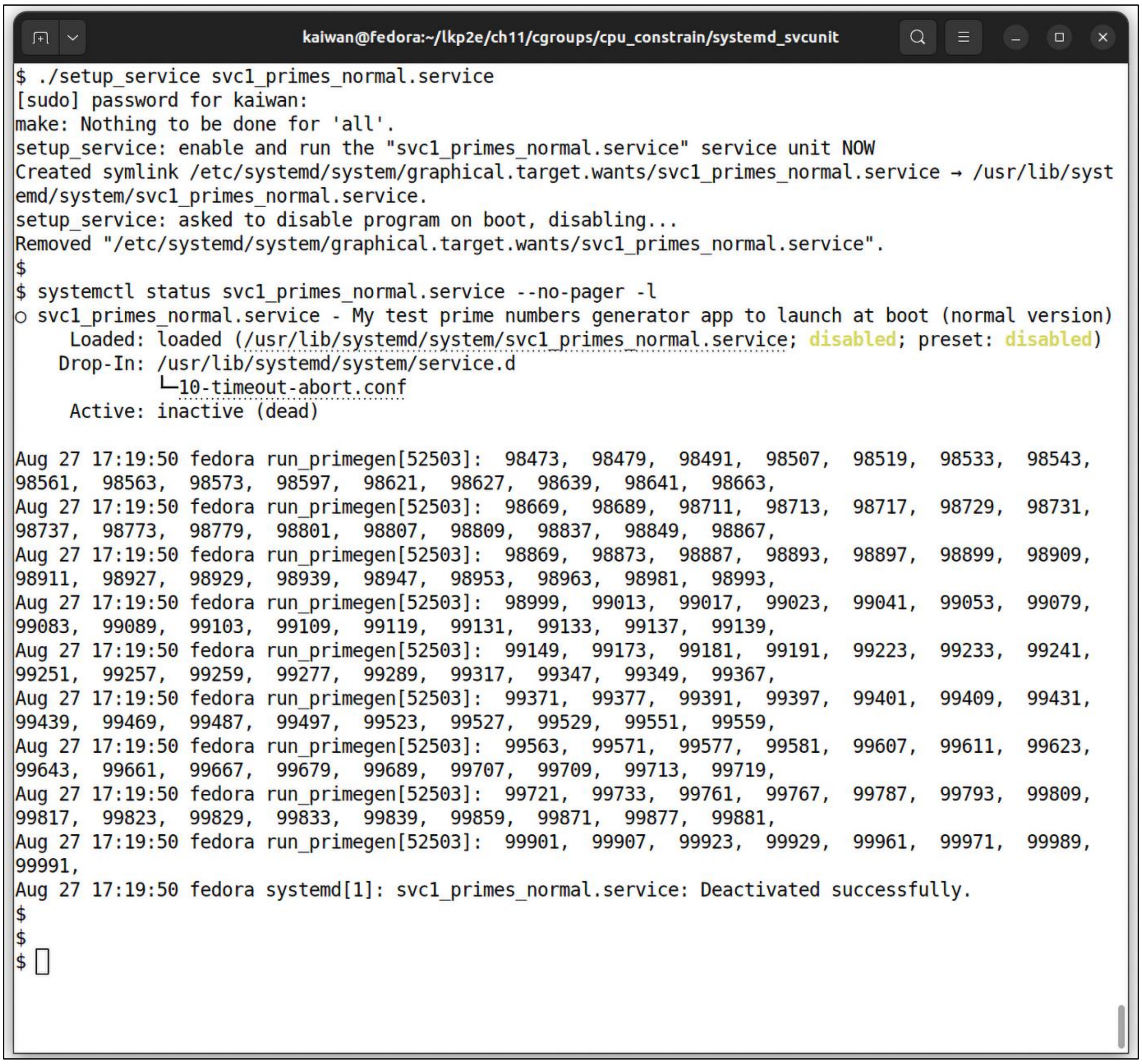
Nice=
Sets the default nice level (scheduling priority) for executed processes. Takes an integer between -20 (highest priority) and 19 (lowest priority). In case of resource contention, smaller values mean more resources will be made available to the unit's processes, larger values mean less resources will be made available. See setpriority(2) for details.

CPUSchedulingPolicy=
Sets the CPU scheduling policy for executed processes. Takes one of **other**, **batch**, **idle**, **fifo** or **rr**. See sched_setscheduler(2) for details.

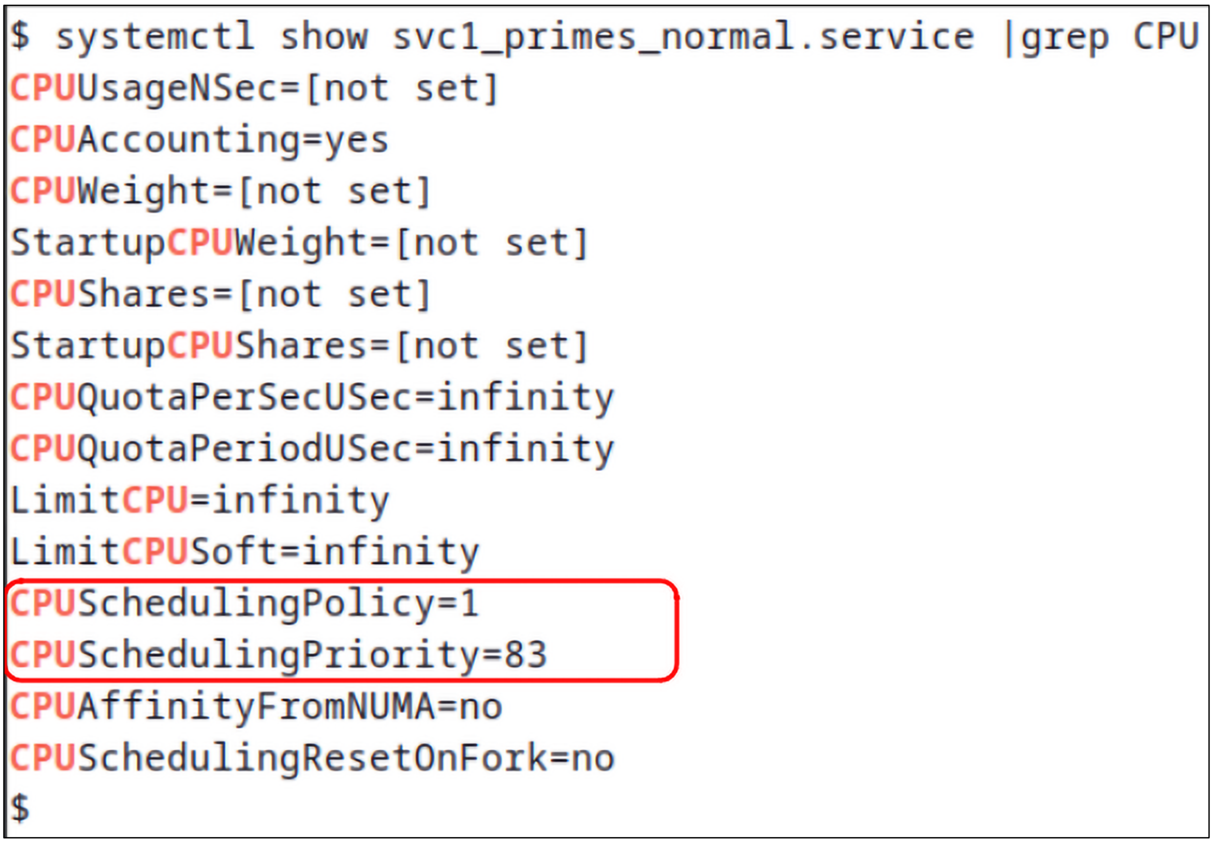
CPUSchedulingPriority=
Sets the CPU scheduling priority for executed processes. The available priority range depends on the selected CPU scheduling policy (see above). For real-time scheduling policies an integer between 1 (lowest priority) and 99 (highest priority) can be used. In case of CPU resource contention, smaller values mean less CPU time is made available to the service, larger values mean more. See sched_setscheduler(2) for details.

CPUSchedulingResetOnFork=
Takes a boolean argument. If true, elevated CPU scheduling priorities and policies will be reset when the executed processes call fork(2), and can hence not leak into child processes. See sched_setscheduler(2) for details. Defaults to false.

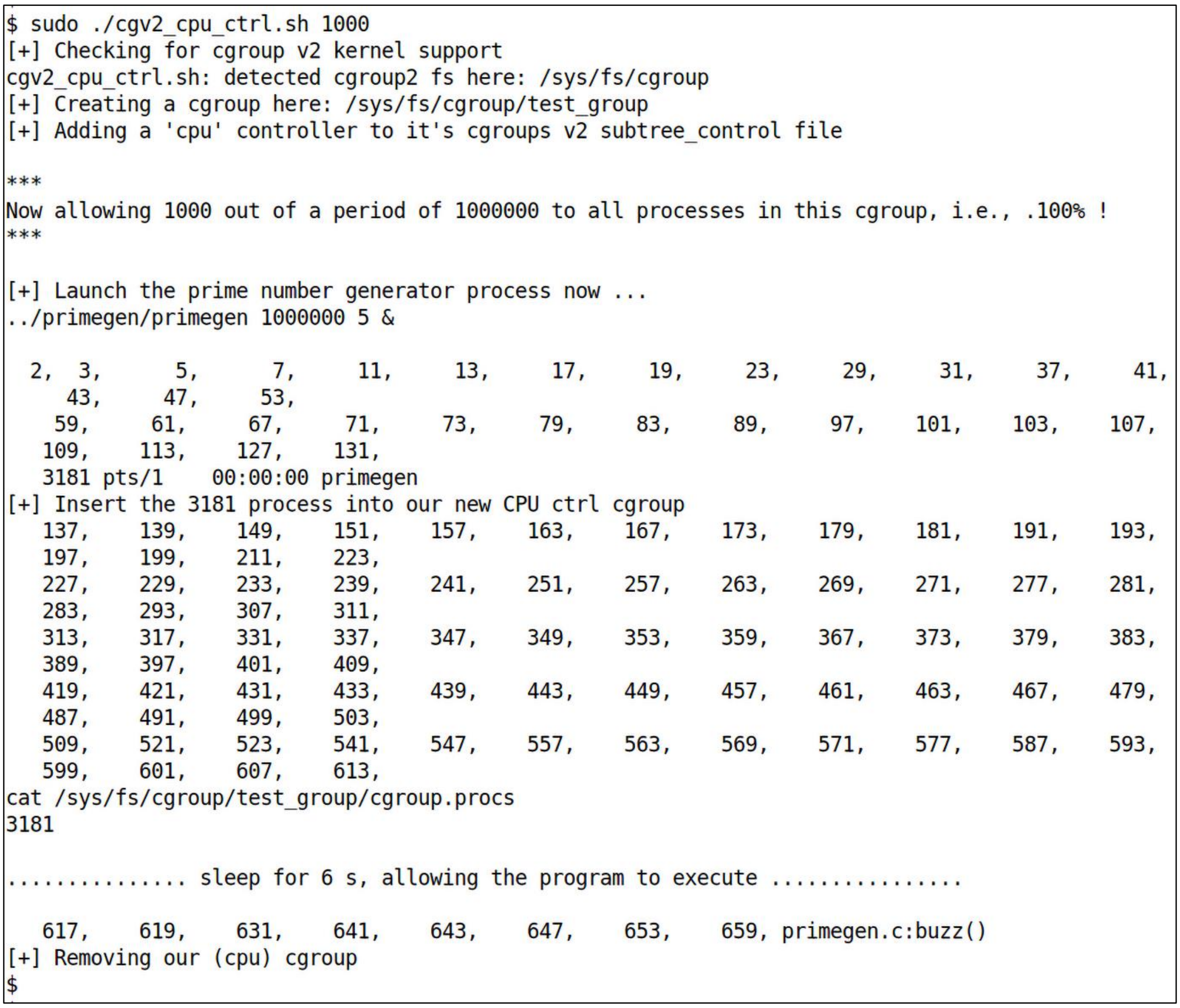
CPUAffinity=
Controls the CPU affinity of the executed processes. Takes a list of CPU indices or ranges separated by either whitespace or commas. Alternatively, takes a special "numa" value in which case systemd automatically derives allowed CPU range based on the value of *NUMAMask=* option. CPU ranges are specified by the lower and upper CPU indices separated by a dash. This option may be specified more than once, in which case the specified CPU affinity masks are merged. If the empty string is assigned, the mask is reset, all assignments prior to this will have no effect. See sched_setaffinity(2) for details.


kaiwan@fedora:~/lkp2e/ch11/cgroups/cpu_constrain/systemd_svcunit

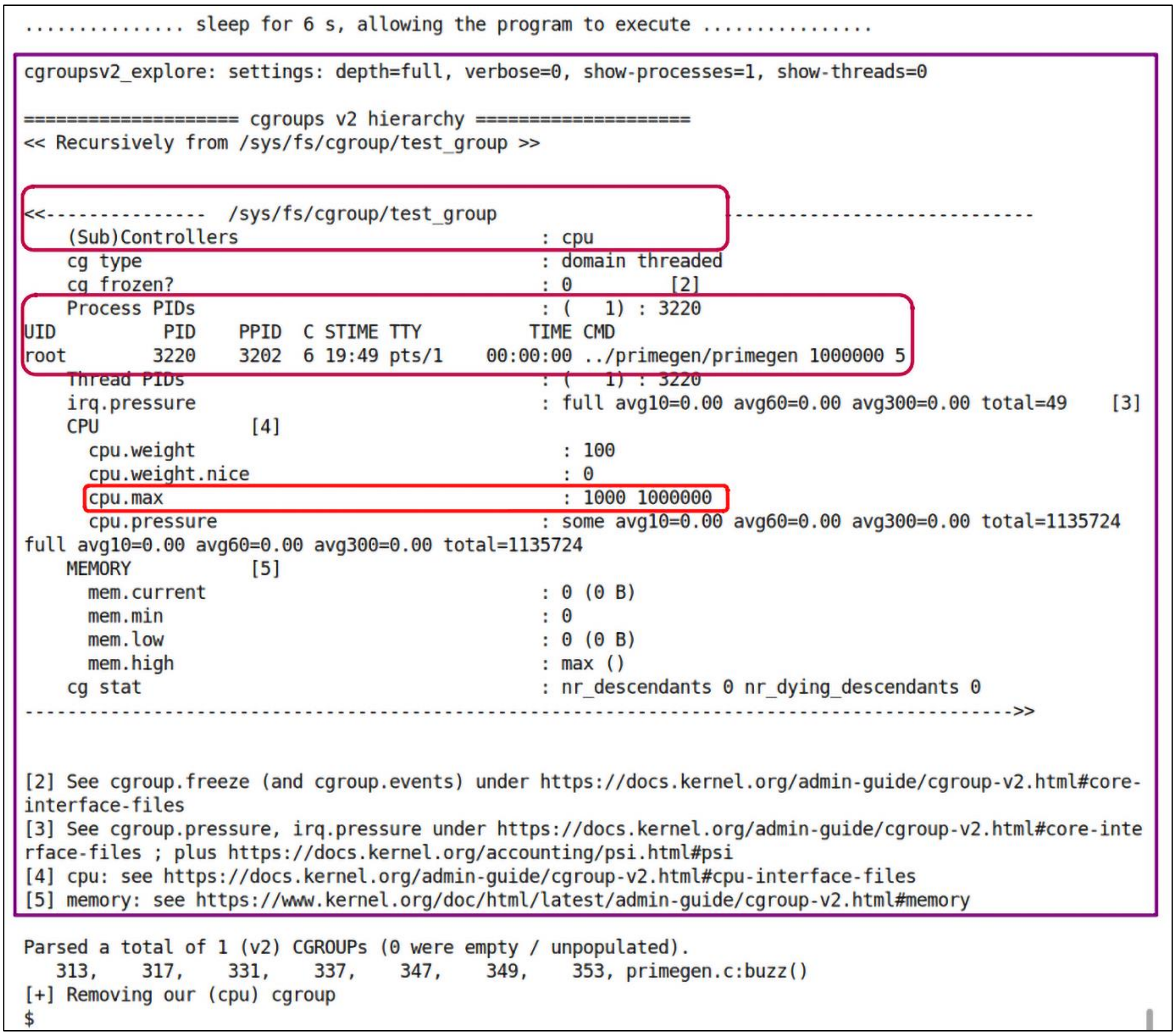
```
$ ./setup_service svc1_primes_normal.service
[sudo] password for kaiwan:
make: Nothing to be done for 'all'.
setup_service: enable and run the "svc1_primes_normal.service" service unit NOW
Created symlink /etc/systemd/system/graphical.target.wants/svc1_primes_normal.service → /usr/lib/syst
emd/system/svc1_primes_normal.service.
setup_service: asked to disable program on boot, disabling...
Removed "/etc/systemd/system/graphical.target.wants/svc1_primes_normal.service".
$
$ systemctl status svc1_primes_normal.service --no-pager -l
○ svc1_primes_normal.service - My test prime numbers generator app to launch at boot (normal version)
     Loaded: loaded (/usr/lib/systemd/system/svc1_primes_normal.service; disabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/service.d
             └─10-timeout-abort.conf
     Active: inactive (dead)

Aug 27 17:19:50 fedora run_primegen[52503]:  98473,  98479,  98491,  98507,  98519,  98533,  98543,
98561,  98563,  98573,  98597,  98621,  98627,  98639,  98641,  98663,
Aug 27 17:19:50 fedora run_primegen[52503]:  98669,  98689,  98711,  98713,  98717,  98729,  98731,
98737,  98773,  98779,  98801,  98807,  98809,  98837,  98849,  98867,
Aug 27 17:19:50 fedora run_primegen[52503]:  98869,  98873,  98887,  98893,  98897,  98899,  98909,
98911,  98927,  98929,  98939,  98947,  98953,  98963,  98981,  98993,
Aug 27 17:19:50 fedora run_primegen[52503]:  98999,  99013,  99017,  99023,  99041,  99053,  99079,
99083,  99089,  99103,  99109,  99119,  99131,  99133,  99137,  99139,
Aug 27 17:19:50 fedora run_primegen[52503]:  99149,  99173,  99181,  99191,  99223,  99233,  99241,
99251,  99257,  99259,  99277,  99289,  99317,  99347,  99349,  99367,
Aug 27 17:19:50 fedora run_primegen[52503]:  99371,  99377,  99391,  99397,  99401,  99409,  99431,
99439,  99469,  99487,  99497,  99523,  99527,  99529,  99551,  99559,
Aug 27 17:19:50 fedora run_primegen[52503]:  99563,  99571,  99577,  99581,  99607,  99611,  99623,
99643,  99661,  99667,  99679,  99689,  99707,  99709,  99713,  99719,
Aug 27 17:19:50 fedora run_primegen[52503]:  99721,  99733,  99761,  99767,  99787,  99793,  99809,
99817,  99823,  99829,  99833,  99839,  99859,  99871,  99877,  99881,
Aug 27 17:19:50 fedora run_primegen[52503]:  99901,  99907,  99923,  99929,  99961,  99971,  99989,
99991,
Aug 27 17:19:50 fedora systemd[1]: svc1_primes_normal.service: Deactivated successfully.
$
$
$ 
```



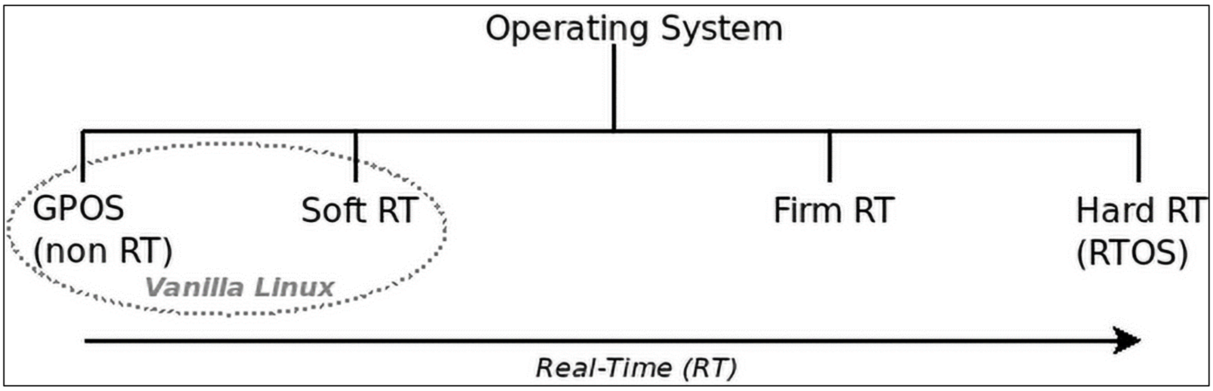
```
$ systemctl show svc1_primes_normal.service |grep CPU
CPUUsageNSec=[not set]
CPUAccounting=yes
CPUWeight=[not set]
StartupCPUWeight=[not set]
CPUShares=[not set]
StartupCPUShares=[not set]
CPUQuotaPerSecUSec=infinity
CPUQuotaPeriodUSec=infinity
LimitCPU=infinity
LimitCPUSoft=infinity
CPUSchedulingPolicy=1
CPUSchedulingPriority=83
CPUAffinityFromNUMA=no
CPUSchedulingResetOnFork=no
$
```



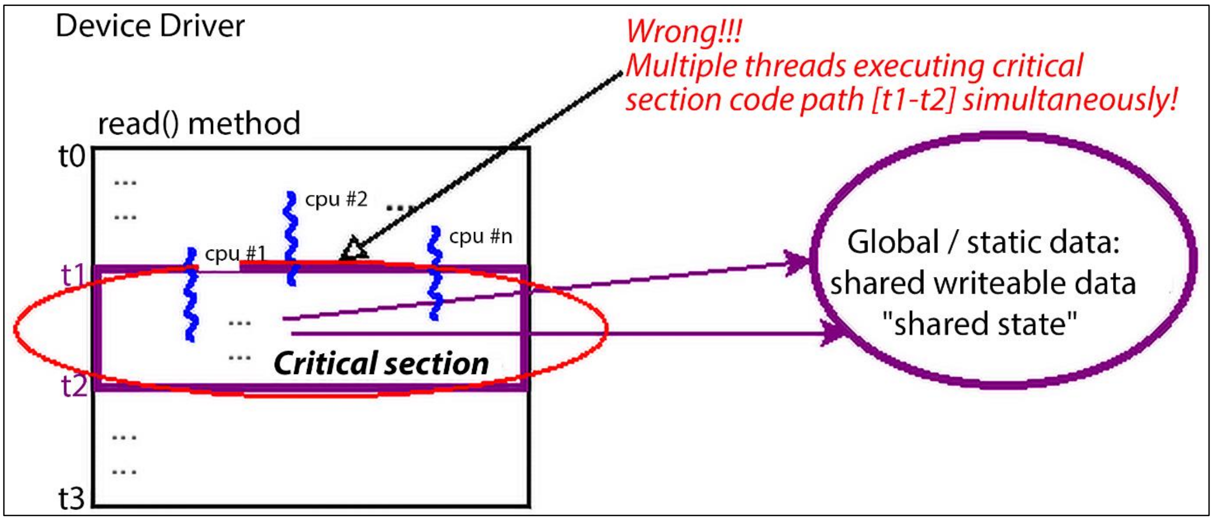
```
$ sudo ./cgv2_cpu_ctrl.sh 1000
[+] Checking for cgroup v2 kernel support
cgv2_cpu_ctrl.sh: detected cgroup2 fs here: /sys/fs/cgroup
[+] Creating a cgroup here: /sys/fs/cgroup/test_group
[+] Adding a 'cpu' controller to it's cgroups v2 subtree_control file

***
Now allowing 1000 out of a period of 1000000 to all processes in this cgroup, i.e., .100% !
***

[+] Launch the prime number generator process now ...
../primegen/primegen 1000000 5 &

  2,  3,      5,      7,     11,     13,     17,     19,     23,     29,     31,     37,     41,
     43,     47,     53,
    59,      61,      67,      71,      73,      79,      83,      89,      97,     101,     103,     107,
   109,     113,     127,     131,
   3181 pts/1    00:00:00 primegen
[+] Insert the 3181 process into our new CPU ctrl cgroup
   137,     139,     149,     151,     157,     163,     167,     173,     179,     181,     191,     193,
   197,     199,     211,     223,
   227,     229,     233,     239,     241,     251,     257,     263,     269,     271,     277,     281,
   283,     293,     307,     311,
   313,     317,     331,     337,     347,     349,     353,     359,     367,     373,     379,     383,
   389,     397,     401,     409,
   419,     421,     431,     433,     439,     443,     449,     457,     461,     463,     467,     479,
   487,     491,     499,     503,
   509,     521,     523,     541,     547,     557,     563,     569,     571,     577,     587,     593,
   599,     601,     607,     613,
cat /sys/fs/cgroup/test_group/cgroup.procs
3181

............... sleep for 6 s, allowing the program to execute ................

   617,     619,     631,     641,     643,     647,     653,     659, primegen.c:buzz()
[+] Removing our (cpu) cgroup
$
```



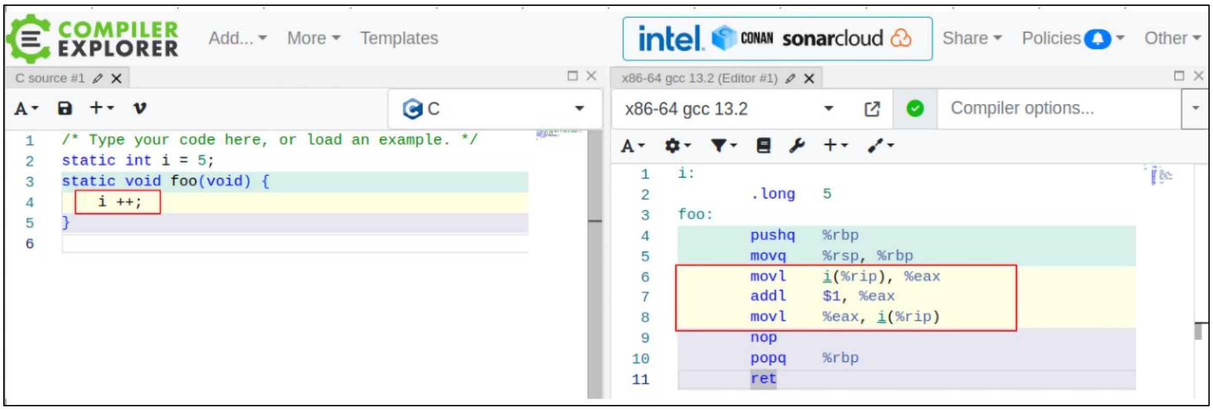
```
............... sleep for 6 s, allowing the program to execute ................

cgroupsv2_explore: settings: depth=full, verbose=0, show-processes=1, show-threads=0

==================== cgroups v2 hierarchy ====================
<< Recursively from /sys/fs/cgroup/test_group >>


<<---------------  /sys/fs/cgroup/test_group                    -----------------------------
    (Sub)Controllers                                   : cpu
    cg type                                            : domain threaded
    cg frozen?                                         : 0          [2]
    Process PIDs                                       : (    1) : 3220
UID          PID    PPID  C STIME TTY          TIME CMD
root        3220    3202  6 19:49 pts/1    00:00:00 ../primegen/primegen 1000000 5
    Thread PIDs                                        : (    1) : 3220
    irq.pressure                                       : full avg10=0.00 avg60=0.00 avg300=0.00 total=49     [3]
    CPU                [4]
      cpu.weight                                         : 100
      cpu.weight.nice                                    : 0
      cpu.max                                            : 1000 1000000
      cpu.pressure                                     : some avg10=0.00 avg60=0.00 avg300=0.00 total=1135724
full avg10=0.00 avg60=0.00 avg300=0.00 total=1135724
    MEMORY             [5]
      mem.current                                      : 0 (0 B)
      mem.min                                          : 0
      mem.low                                          : 0 (0 B)
      mem.high                                         : max ()
    cg stat                                            : nr_descendants 0 nr_dying_descendants 0
----------------------------------------------------------------------------------------------->>


[2] See cgroup.freeze (and cgroup.events) under https://docs.kernel.org/admin-guide/cgroup-v2.html#core-
interface-files
[3] See cgroup.pressure, irq.pressure under https://docs.kernel.org/admin-guide/cgroup-v2.html#core-inte
rface-files ; plus https://docs.kernel.org/accounting/psi.html#psi
[4] cpu: see https://docs.kernel.org/admin-guide/cgroup-v2.html#cpu-interface-files
[5] memory: see https://www.kernel.org/doc/html/latest/admin-guide/cgroup-v2.html#memory

Parsed a total of 1 (v2) CGROUPs (0 were empty / unpopulated).
   313,    317,    331,    337,    347,    349,    353, primegen.c:buzz()
[+] Removing our (cpu) cgroup
$
```



Chapter 12: Kernel Synchronization - Part 1

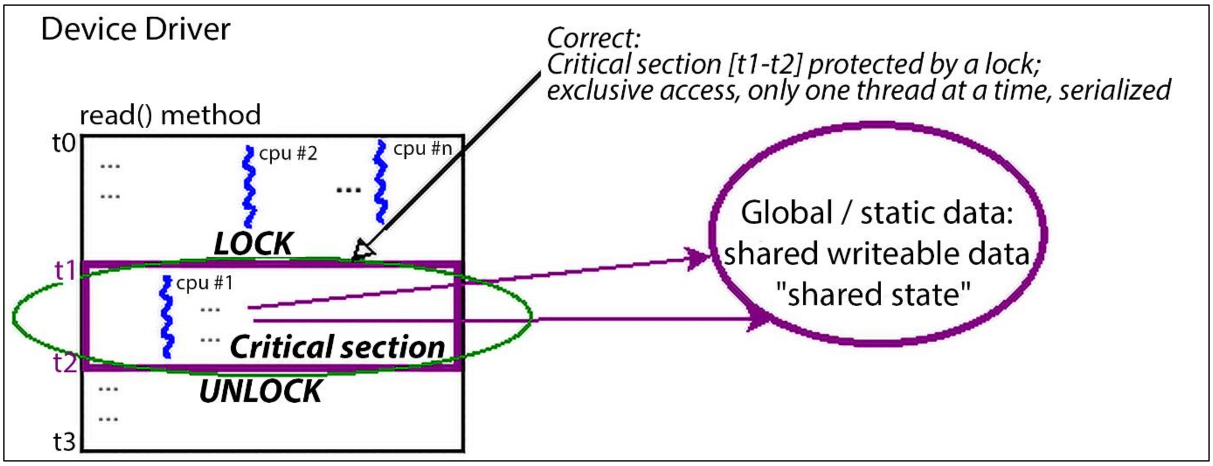


Wrong!!!
Multiple threads executing critical section code path [t1-t2] simultaneously!

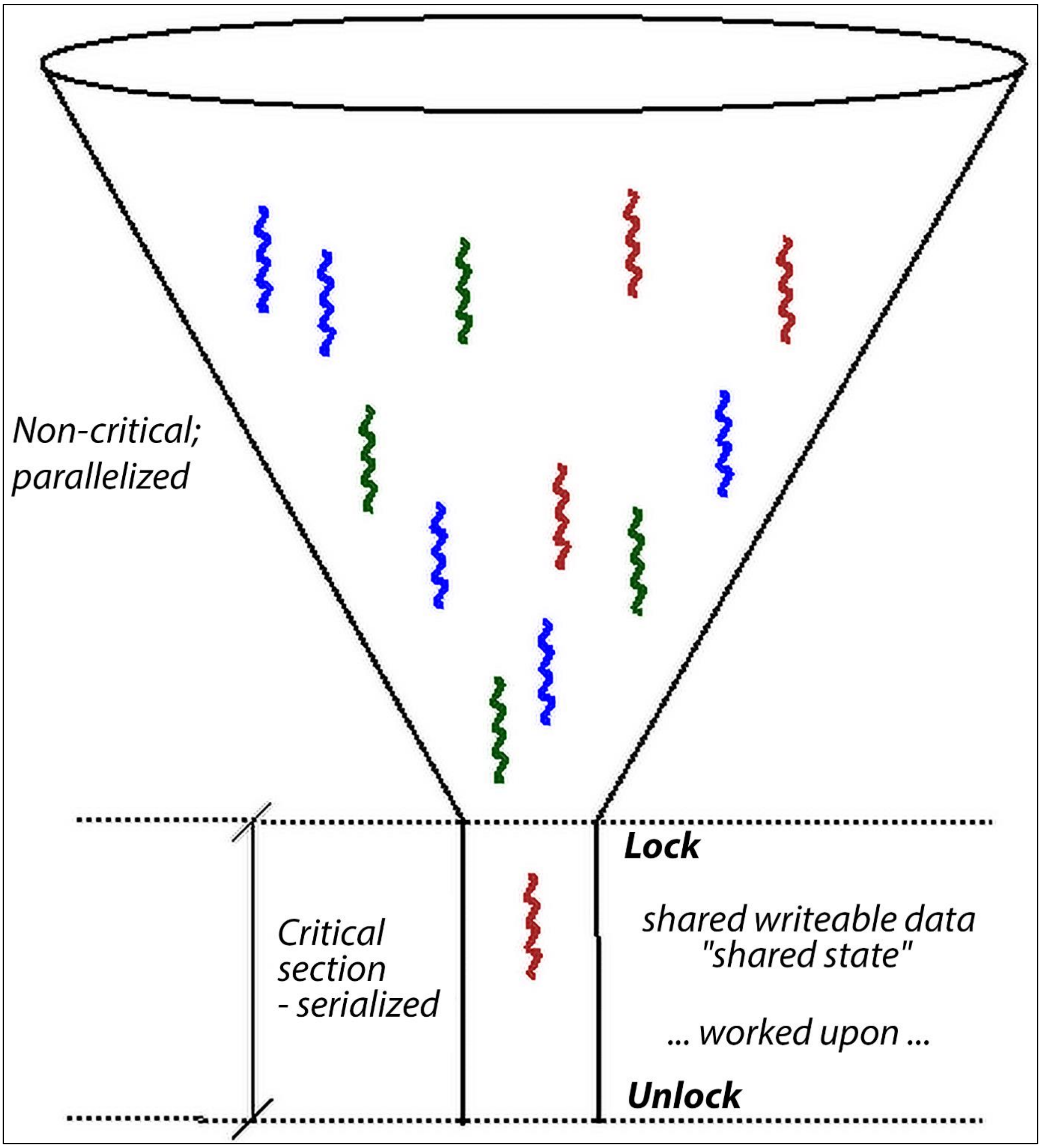


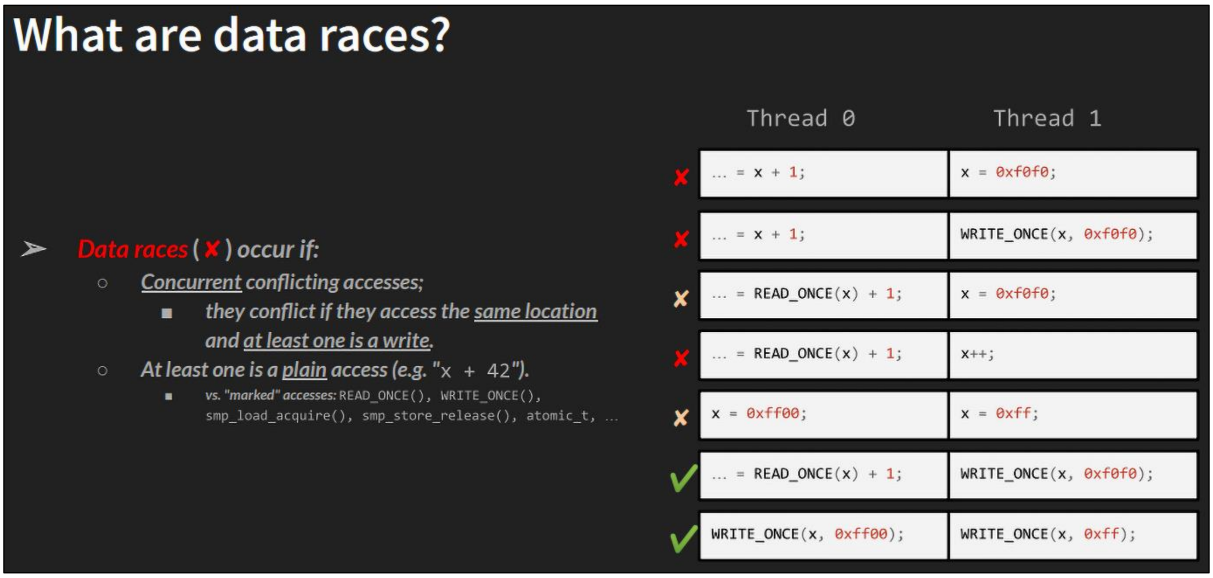
```c
/* Type your code here, or load an example. */
static int i = 5;
static void foo(void) {
    i ++;
}
```

```
i:
        .long   5
foo:
        pushq   %rbp
        movq    %rsp, %rbp
        movl    i(%rip), %eax
        addl    $1, %eax
        movl    %eax, i(%rip)
        nop
        popq    %rbp
        ret
```



Correct:
Critical section [t1-t2] protected by a lock; exclusive access, only one thread at a time, serialized

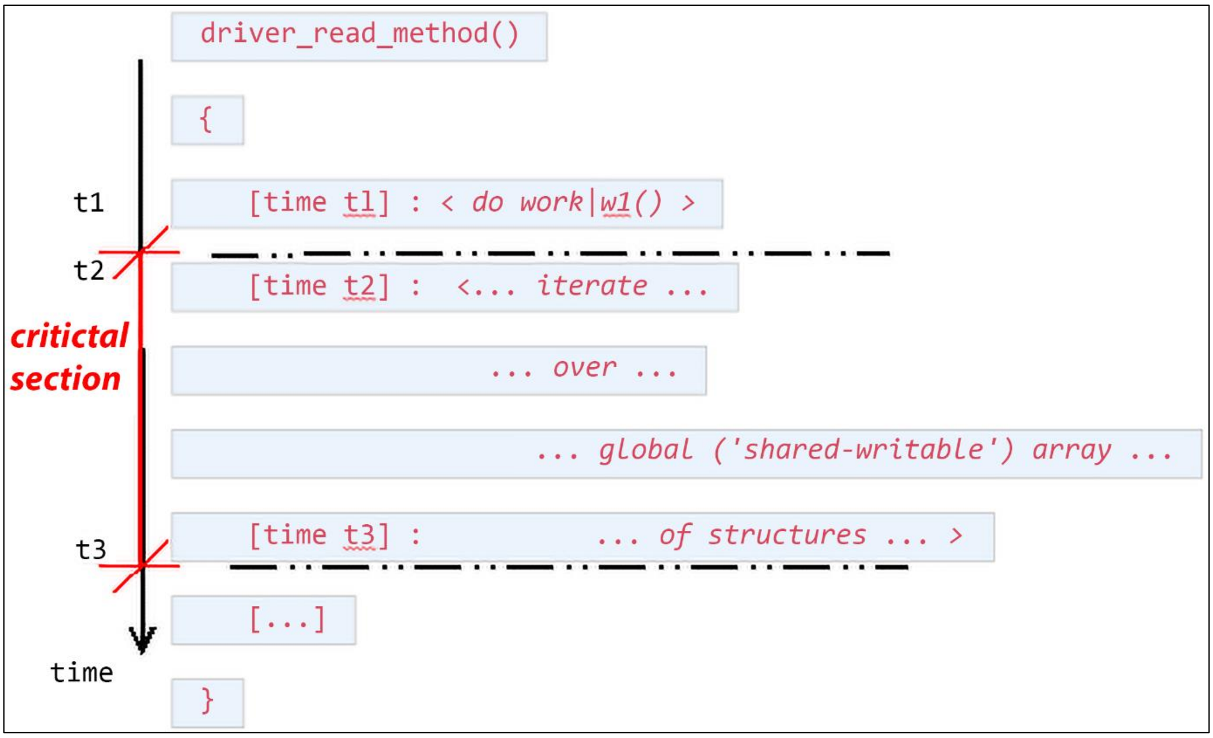

Non-critical;
parallelized
Lock
Critical
section
- serialized
shared writeable data
"shared state"
... worked upon ...
Unlock



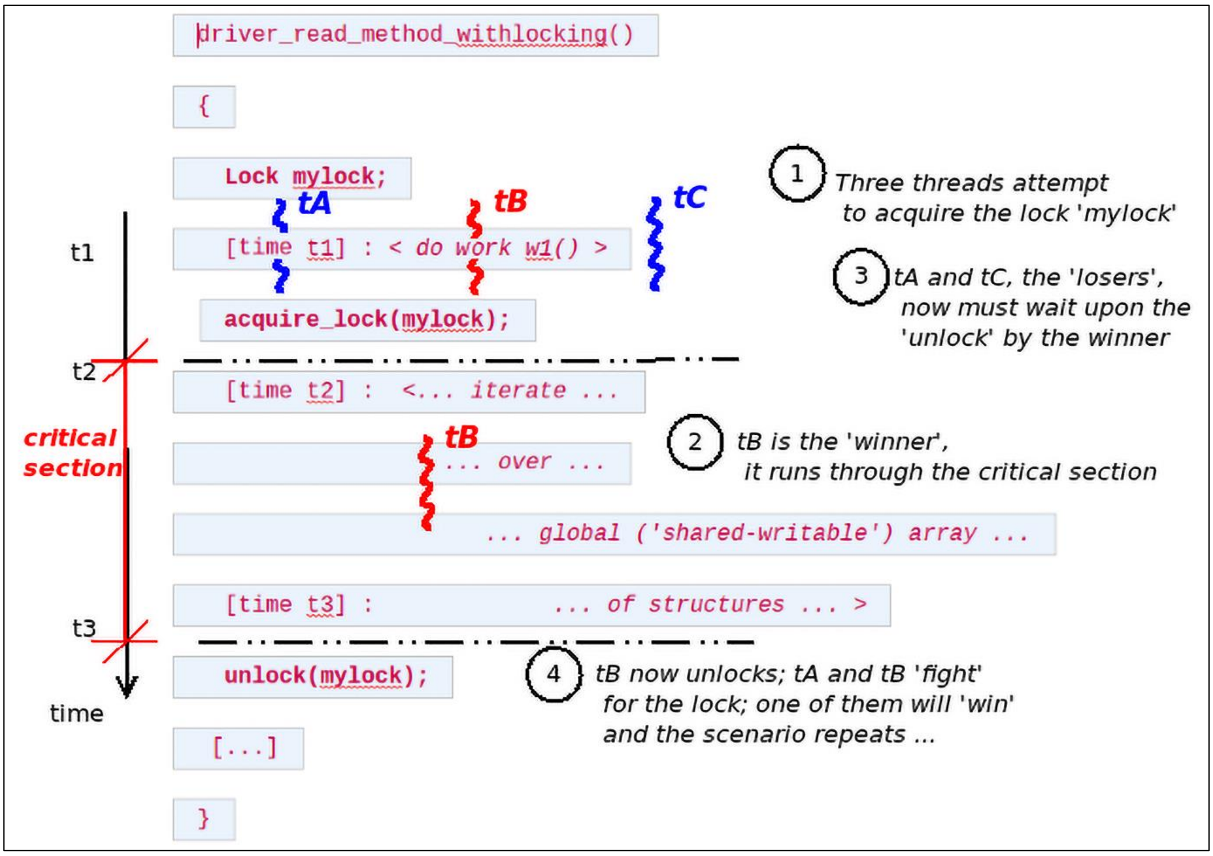
What are data races?

- ➢ *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);`



```
driver_read_method()
{
    [time t1] : < do work|w1() >
    [time t2] :  <... iterate ...
                          ... over ...
                              ... global ('shared-writable') array ...
    [time t3] :               ... of structures ... >
    [...]
}
```



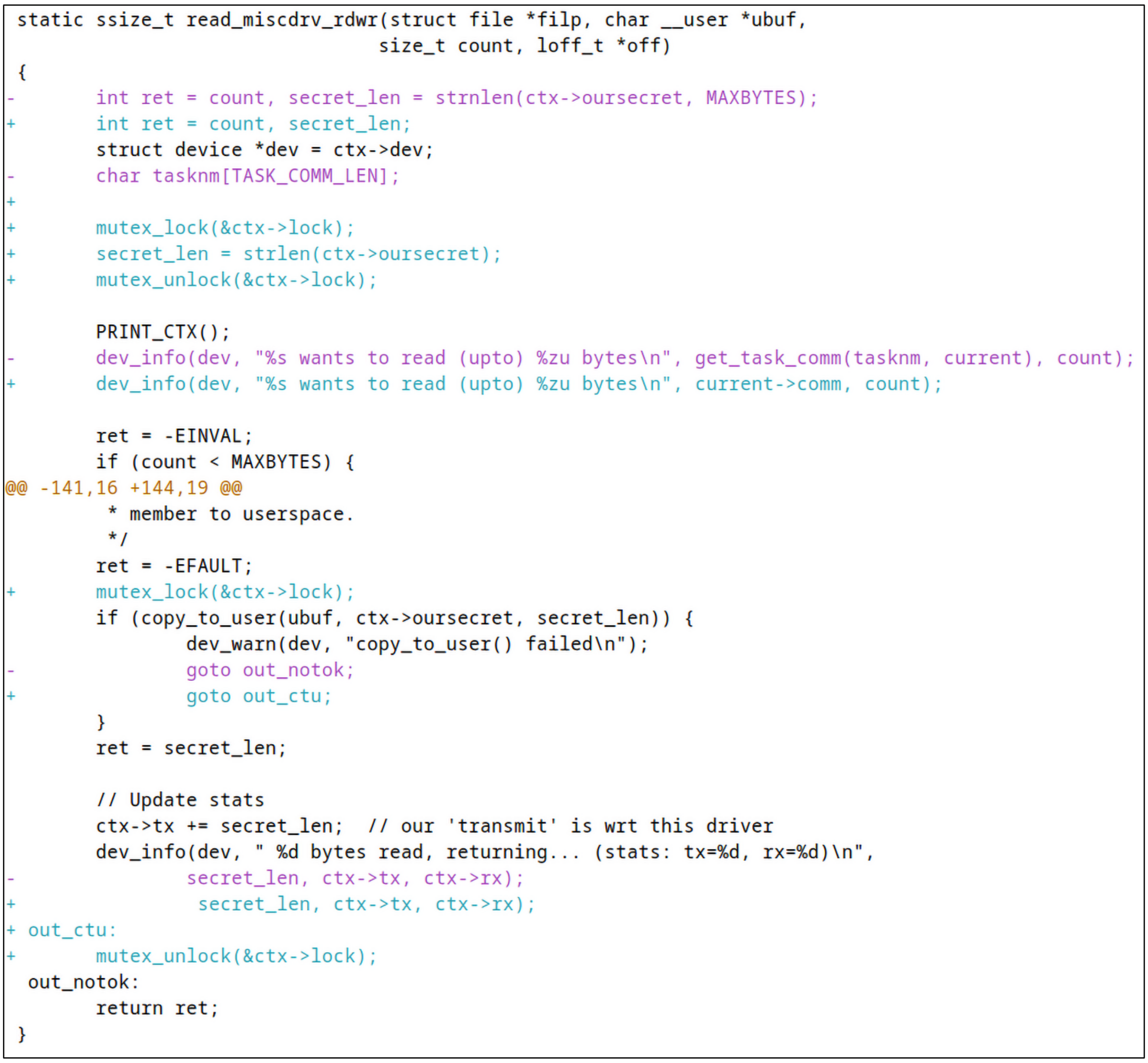
```
driver_read_method_withlocking()
{
    Lock mylock;
    [time t1] : < do work w1() >
    acquire_lock(mylock);
    [time t2] :  <... iterate ...
                ... over ...
                      ... global ('shared-writable') array ...
    [time t3] :              ... of structures ... >
    unlock(mylock);
  [...]
}
```

1 Three threads attempt to acquire the lock 'mylock'

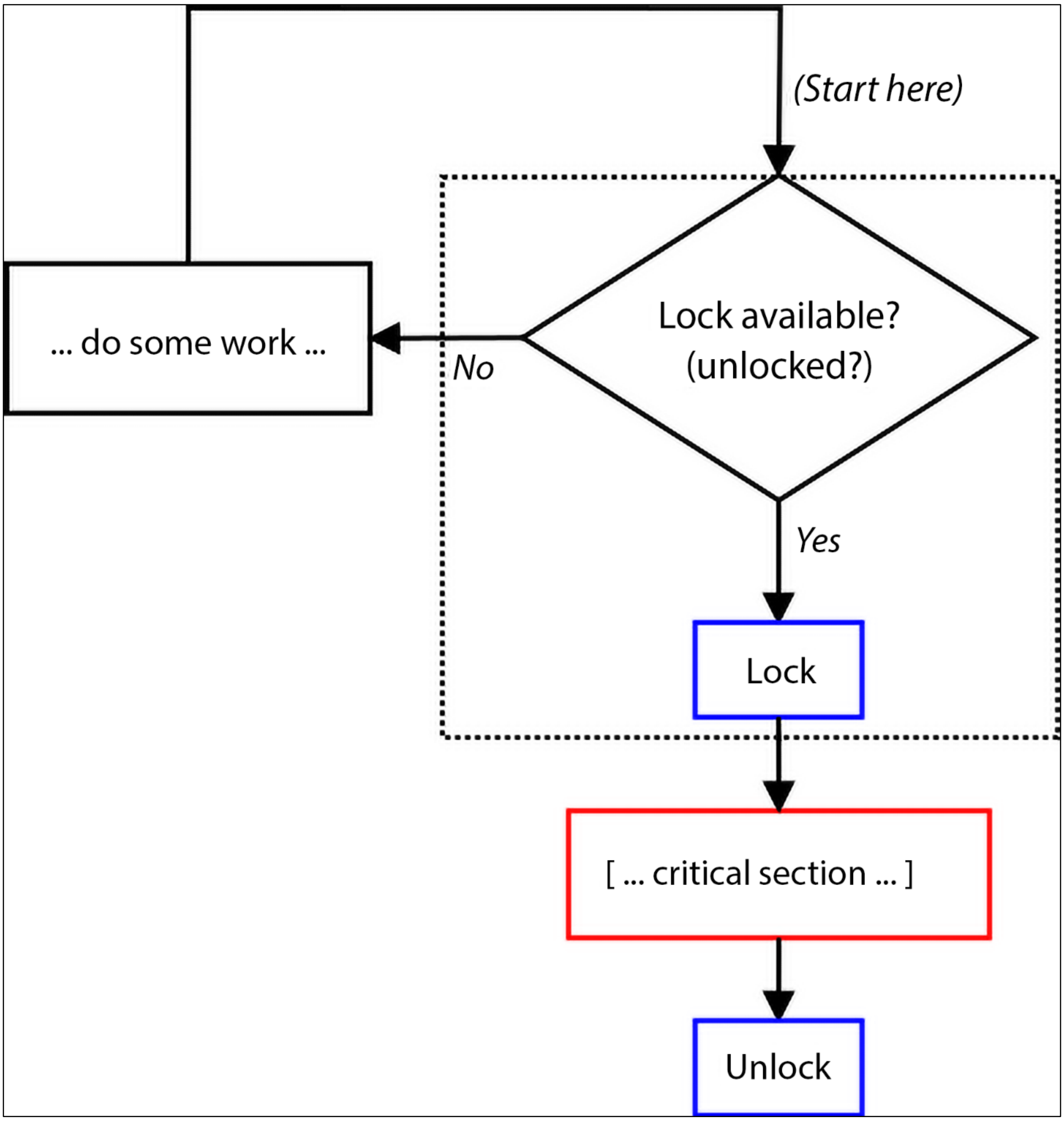
2 tB is the 'winner', it runs through the critical section

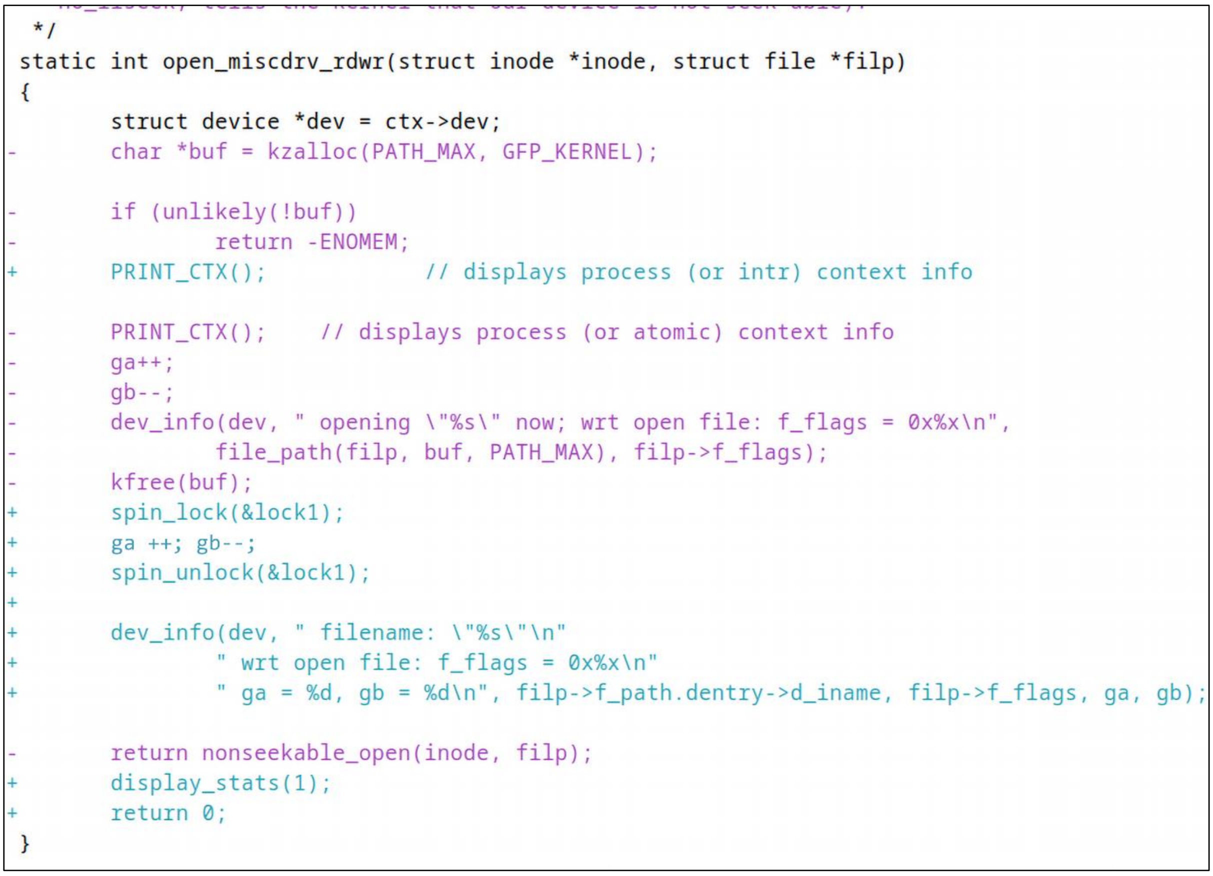
3 tA and tC, the 'losers', now must wait upon the 'unlock' by the winner

4 tB now unlocks; tA and tB 'fight' for the lock; one of them will 'win' and the scenario repeats ...

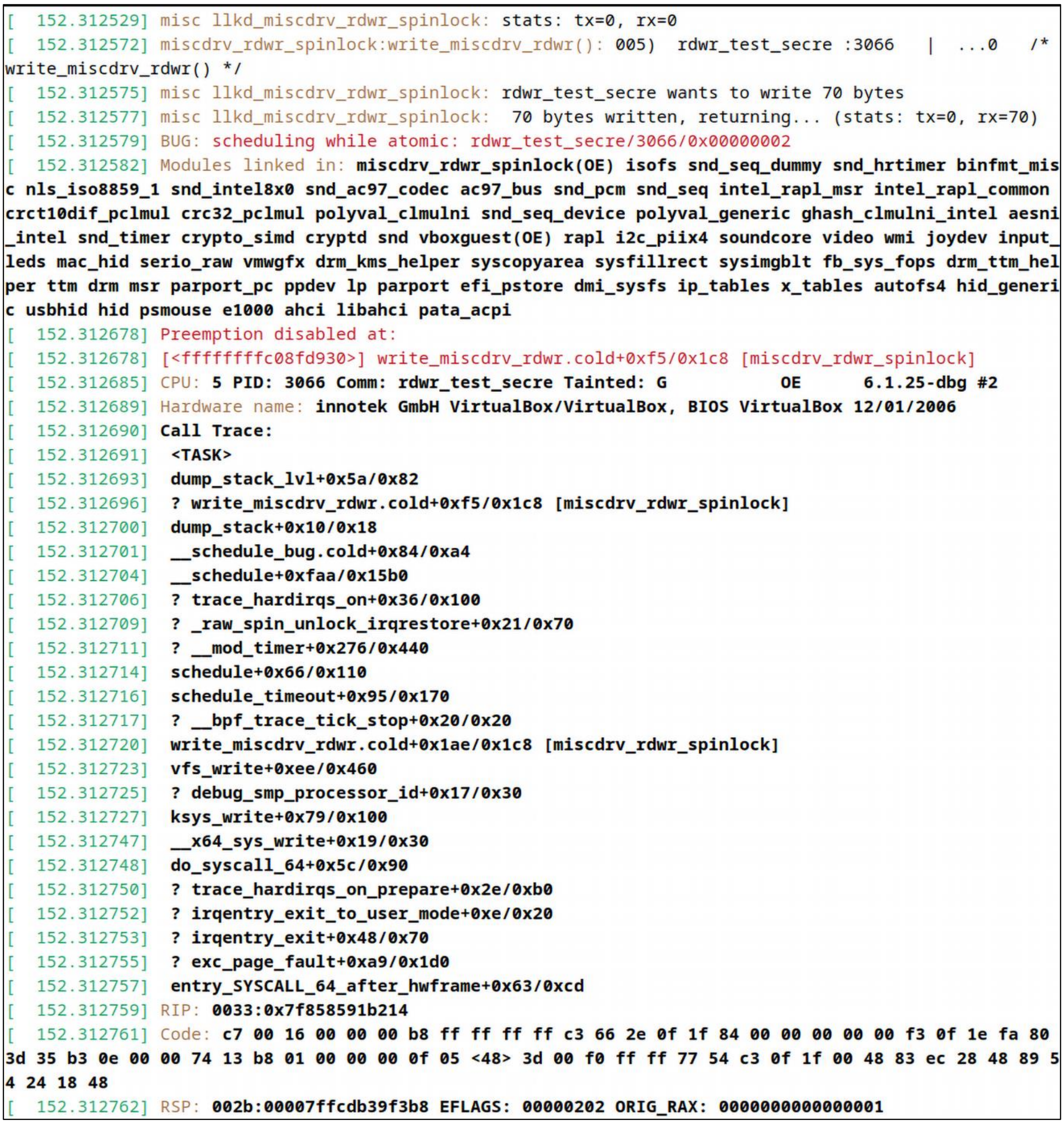


```
 static ssize_t read_miscdrv_rdwr(struct file *filp, char __user *ubuf,
                                 size_t count, loff_t *off)
 {
-       int ret = count, secret_len = strnlen(ctx->oursecret, MAXBYTES);
+       int ret = count, secret_len;
        struct device *dev = ctx->dev;
-       char tasknm[TASK_COMM_LEN];
+
+       mutex_lock(&ctx->lock);
+       secret_len = strlen(ctx->oursecret);
+       mutex_unlock(&ctx->lock);

        PRINT_CTX();
-       dev_info(dev, "%s wants to read (upto) %zu bytes\n", get_task_comm(tasknm, current), count);
+       dev_info(dev, "%s wants to read (upto) %zu bytes\n", current->comm, count);

        ret = -EINVAL;
        if (count < MAXBYTES) {
@@ -141,16 +144,19 @@
         * member to userspace.
         */
        ret = -EFAULT;
+       mutex_lock(&ctx->lock);
        if (copy_to_user(ubuf, ctx->oursecret, secret_len)) {
                dev_warn(dev, "copy_to_user() failed\n");
-               goto out_notok;
+               goto out_ctu;
        }
        ret = secret_len;

        // Update stats
        ctx->tx += secret_len;  // our 'transmit' is wrt this driver
        dev_info(dev, " %d bytes read, returning... (stats: tx=%d, rx=%d)\n",
-               secret_len, ctx->tx, ctx->rx);
+                secret_len, ctx->tx, ctx->rx);
+ out_ctu:
+       mutex_unlock(&ctx->lock);
  out_notok:
        return ret;
 }
```

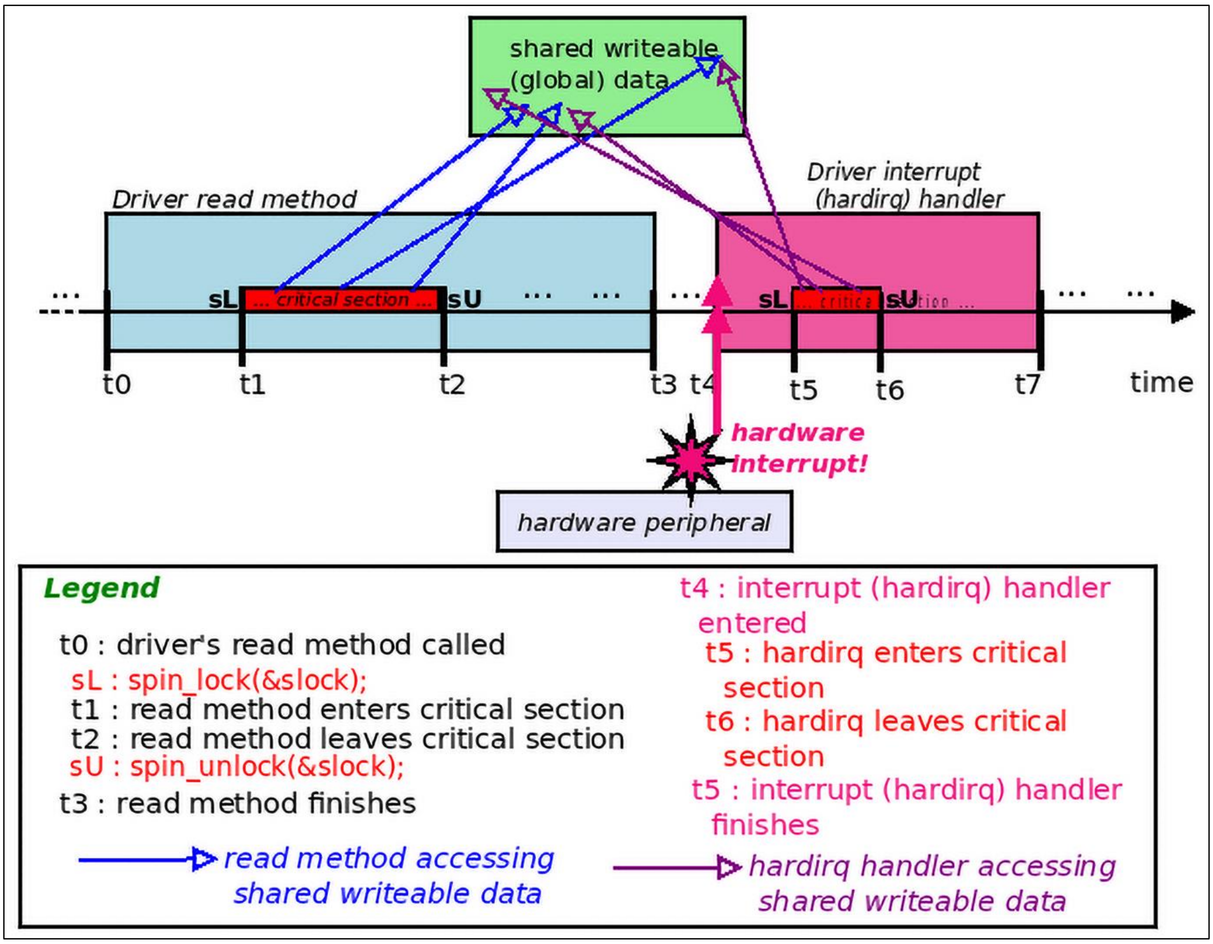

(Start here)
Lock available?
(unlocked?)
No
... do some work ...
Yes
Lock
[... critical section ...]
Unlock



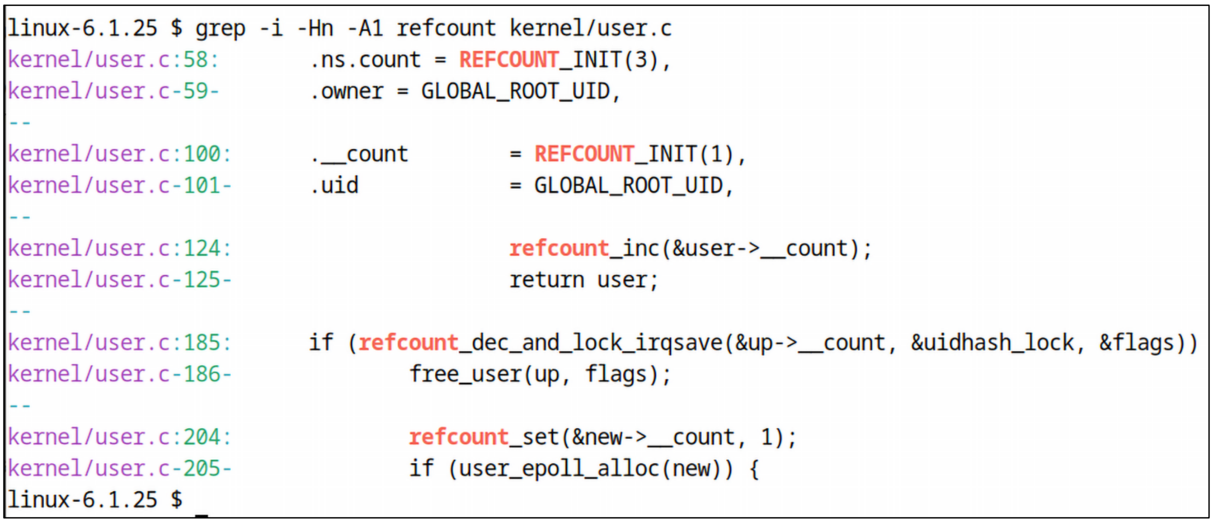
```
   no_llseek; tells the kernel that our device is not seek-able).
  */
 static int open_miscdrv_rdwr(struct inode *inode, struct file *filp)
 {
         struct device *dev = ctx->dev;
-        char *buf = kzalloc(PATH_MAX, GFP_KERNEL);

-        if (unlikely(!buf))
-                return -ENOMEM;
+        PRINT_CTX();            // displays process (or intr) context info

-        PRINT_CTX();    // displays process (or atomic) context info
-        ga++;
-        gb--;
-        dev_info(dev, " opening \"%s\" now; wrt open file: f_flags = 0x%x\n",
-                file_path(filp, buf, PATH_MAX), filp->f_flags);
-        kfree(buf);
+        spin_lock(&lock1);
+        ga ++; gb--;
+        spin_unlock(&lock1);
+
+        dev_info(dev, " filename: \"%s\"\n"
+                " wrt open file: f_flags = 0x%x\n"
+                " ga = %d, gb = %d\n", filp->f_path.dentry->d_iname, filp->f_flags, ga, gb);

-        return nonseekable_open(inode, filp);
+        display_stats(1);
+        return 0;
 }
```



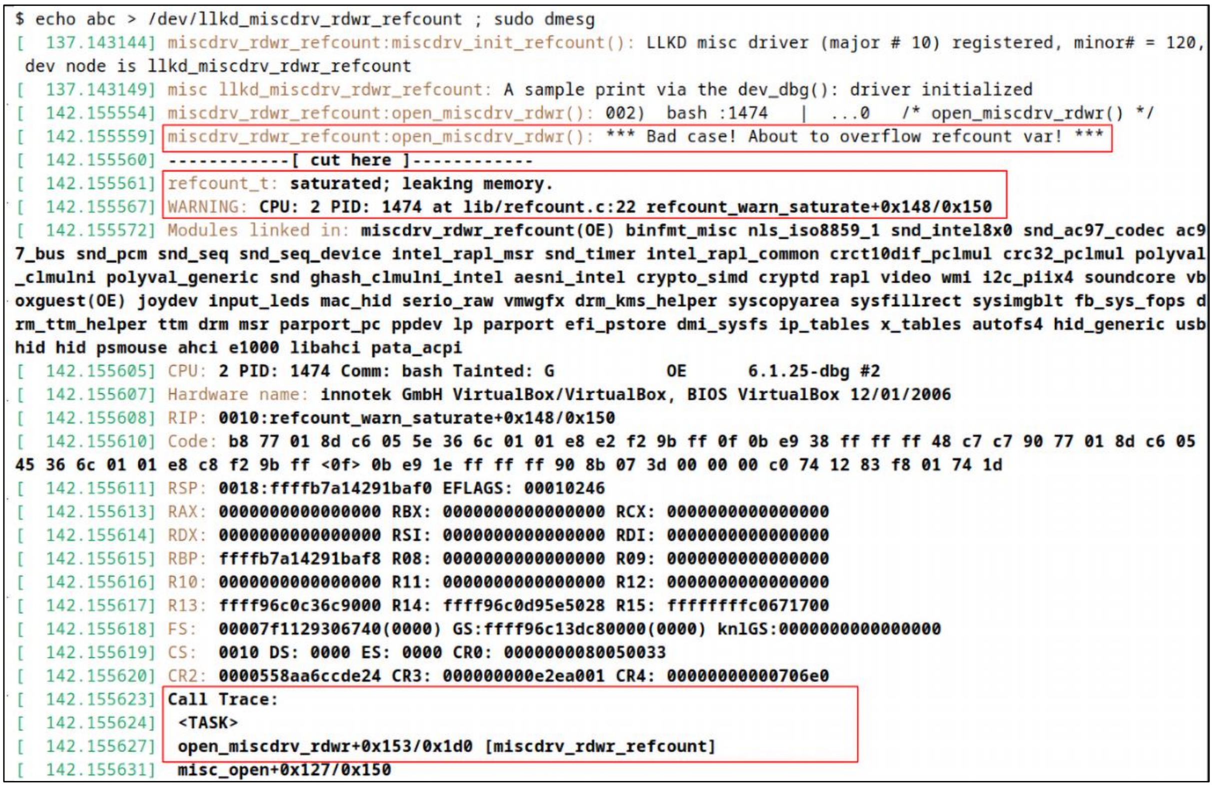
```
[  152.312529] misc llkd_miscdrv_rdwr_spinlock: stats: tx=0, rx=0
[  152.312572] miscdrv_rdwr_spinlock:write_miscdrv_rdwr(): 005)  rdwr_test_secre :3066   |  ...0   /*
write_miscdrv_rdwr() */
[  152.312575] misc llkd_miscdrv_rdwr_spinlock: rdwr_test_secre wants to write 70 bytes
[  152.312577] misc llkd_miscdrv_rdwr_spinlock:  70 bytes written, returning... (stats: tx=0, rx=70)
[  152.312579] BUG: scheduling while atomic: rdwr_test_secre/3066/0x00000002
[  152.312582] Modules linked in: miscdrv_rdwr_spinlock(OE) isofs snd_seq_dummy snd_hrtimer binfmt_mis
c nls_iso8859_1 snd_intel8x0 snd_ac97_codec ac97_bus snd_pcm snd_seq intel_rapl_msr intel_rapl_common
crct10dif_pclmul crc32_pclmul polyval_clmulni snd_seq_device polyval_generic ghash_clmulni_intel aesni
_intel snd_timer crypto_simd cryptd snd vboxguest(OE) rapl i2c_piix4 soundcore video wmi joydev input_
leds mac_hid serio_raw vmwgfx drm_kms_helper syscopyarea sysfillrect sysimgblt fb_sys_fops drm_ttm_hel
per ttm drm msr parport_pc ppdev lp parport efi_pstore dmi_sysfs ip_tables x_tables autofs4 hid_generi
c usbhid hid psmouse e1000 ahci libahci pata_acpi
[  152.312678] Preemption disabled at:
[  152.312678] [<ffffffffc08fd930>] write_miscdrv_rdwr.cold+0xf5/0x1c8 [miscdrv_rdwr_spinlock]
[  152.312685] CPU: 5 PID: 3066 Comm: rdwr_test_secre Tainted: G           OE      6.1.25-dbg #2
[  152.312689] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[  152.312690] Call Trace:
[  152.312691]  <TASK>
[  152.312693]  dump_stack_lvl+0x5a/0x82
[  152.312696]  ? write_miscdrv_rdwr.cold+0xf5/0x1c8 [miscdrv_rdwr_spinlock]
[  152.312700]  dump_stack+0x10/0x18
[  152.312701]  __schedule_bug.cold+0x84/0xa4
[  152.312704]  __schedule+0xfaa/0x15b0
[  152.312706]  ? trace_hardirqs_on+0x36/0x100
[  152.312709]  ? _raw_spin_unlock_irqrestore+0x21/0x70
[  152.312711]  ? __mod_timer+0x276/0x440
[  152.312714]  schedule+0x66/0x110
[  152.312716]  schedule_timeout+0x95/0x170
[  152.312717]  ? __bpf_trace_tick_stop+0x20/0x20
[  152.312720]  write_miscdrv_rdwr.cold+0x1ae/0x1c8 [miscdrv_rdwr_spinlock]
[  152.312723]  vfs_write+0xee/0x460
[  152.312725]  ? debug_smp_processor_id+0x17/0x30
[  152.312727]  ksys_write+0x79/0x100
[  152.312747]  __x64_sys_write+0x19/0x30
[  152.312748]  do_syscall_64+0x5c/0x90
[  152.312750]  ? trace_hardirqs_on_prepare+0x2e/0xb0
[  152.312752]  ? irqentry_exit_to_user_mode+0xe/0x20
[  152.312753]  ? irqentry_exit+0x48/0x70
[  152.312755]  ? exc_page_fault+0xa9/0x1d0
[  152.312757]  entry_SYSCALL_64_after_hwframe+0x63/0xcd
[  152.312759] RIP: 0033:0x7f858591b214
[  152.312761] Code: c7 00 16 00 00 00 b8 ff ff ff ff c3 66 2e 0f 1f 84 00 00 00 00 00 f3 0f 1e fa 80
3d 35 b3 0e 00 00 74 13 b8 01 00 00 00 0f 05 <48> 3d 00 f0 ff ff 77 54 c3 0f 1f 00 48 83 ec 28 48 89 5
4 24 18 48
[  152.312762] RSP: 002b:00007ffcdb39f3b8 EFLAGS: 00000202 ORIG_RAX: 0000000000000001
```


shared writeable (global) data
Driver read method
Driver interrupt (hardirq) handler
sL
... critical section ...
sU
...
sL
... critical section ...
sU
t0
t1
t2
t3
t4
t5
t6
t7
time
hardware interrupt!
hardware peripheral
Legend
t0 : driver's read method called
sL : spin_lock(&slock);
t1 : read method enters critical section
t2 : read method leaves critical section
sU : spin_unlock(&slock);
t3 : read method finishes
read method accessing shared writeable data
t4 : interrupt (hardirq) handler entered
t5 : hardirq enters critical section
t6 : hardirq leaves critical section
t5 : interrupt (hardirq) handler finishes
hardirq handler accessing shared writeable data

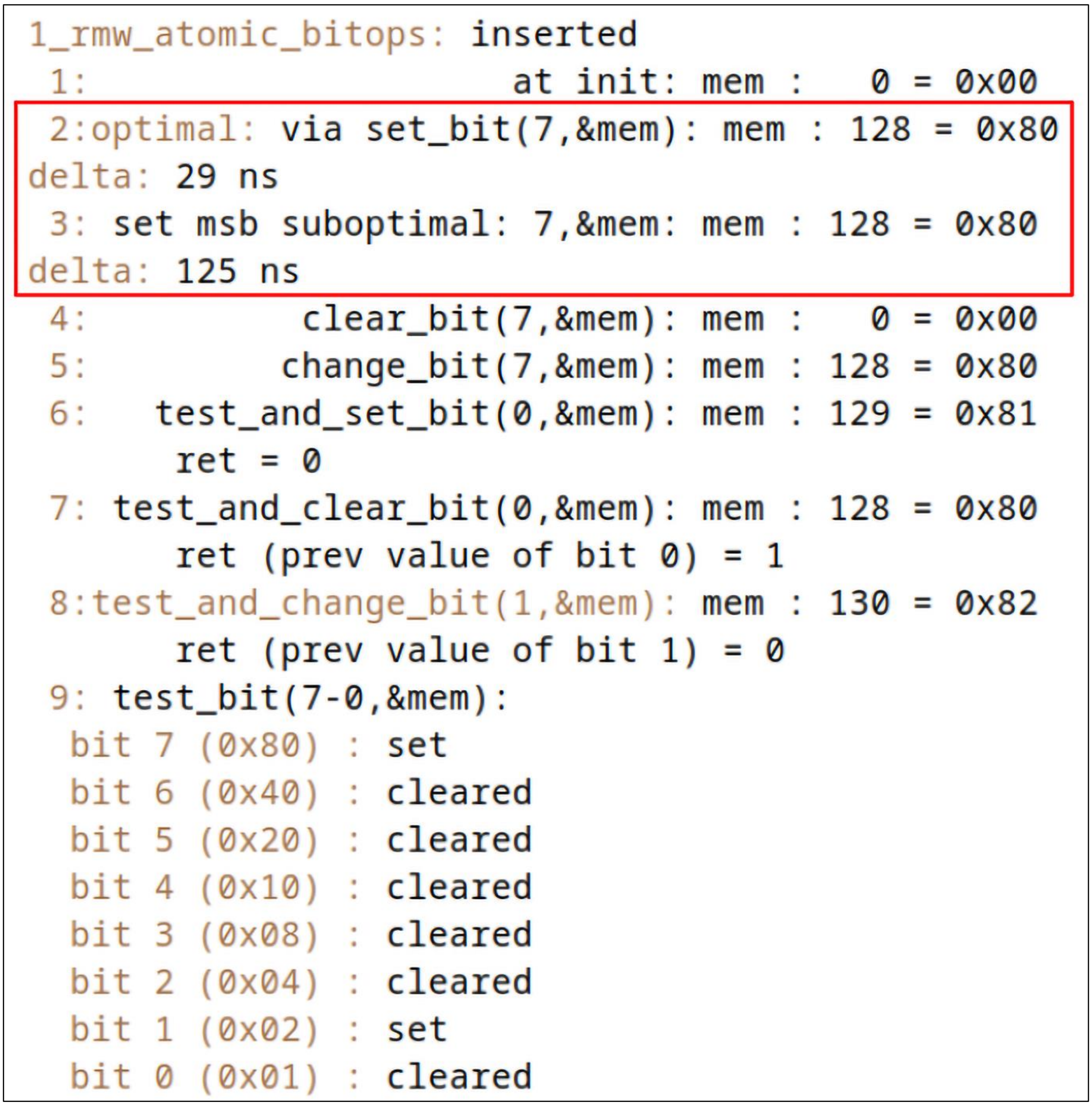
Chapter 13: Kernel Synchronization - Part 2



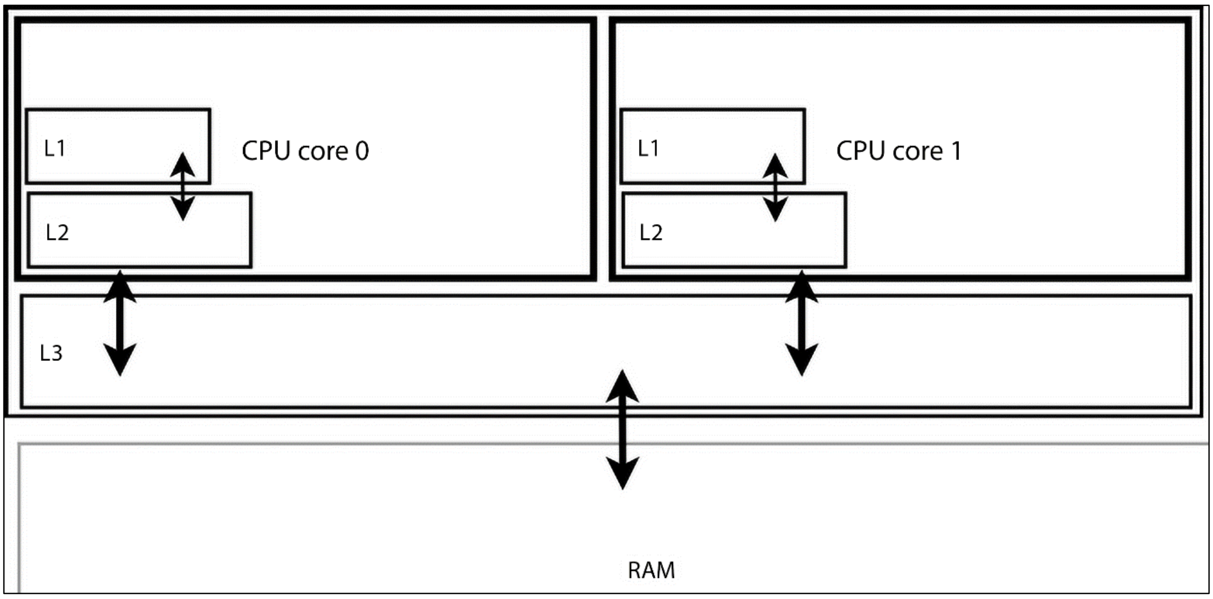
```
linux-6.1.25 $ grep -i -Hn -A1 refcount kernel/user.c
kernel/user.c:58:       .ns.count = REFCOUNT_INIT(3),
kernel/user.c-59-       .owner = GLOBAL_ROOT_UID,
--
kernel/user.c:100:      .__count        = REFCOUNT_INIT(1),
kernel/user.c-101-      .uid            = GLOBAL_ROOT_UID,
--
kernel/user.c:124:                      refcount_inc(&user->__count);
kernel/user.c-125-                      return user;
--
kernel/user.c:185:      if (refcount_dec_and_lock_irqsave(&up->__count, &uidhash_lock, &flags))
kernel/user.c-186-              free_user(up, flags);
--
kernel/user.c:204:              refcount_set(&new->__count, 1);
kernel/user.c-205-              if (user_epoll_alloc(new)) {
linux-6.1.25 $ _
```

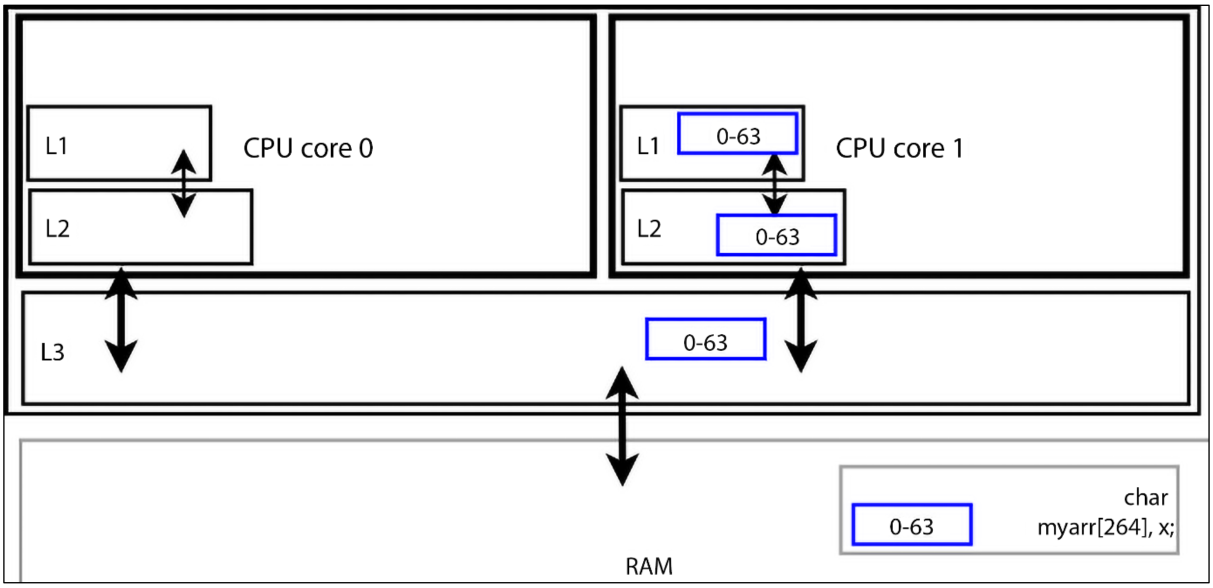


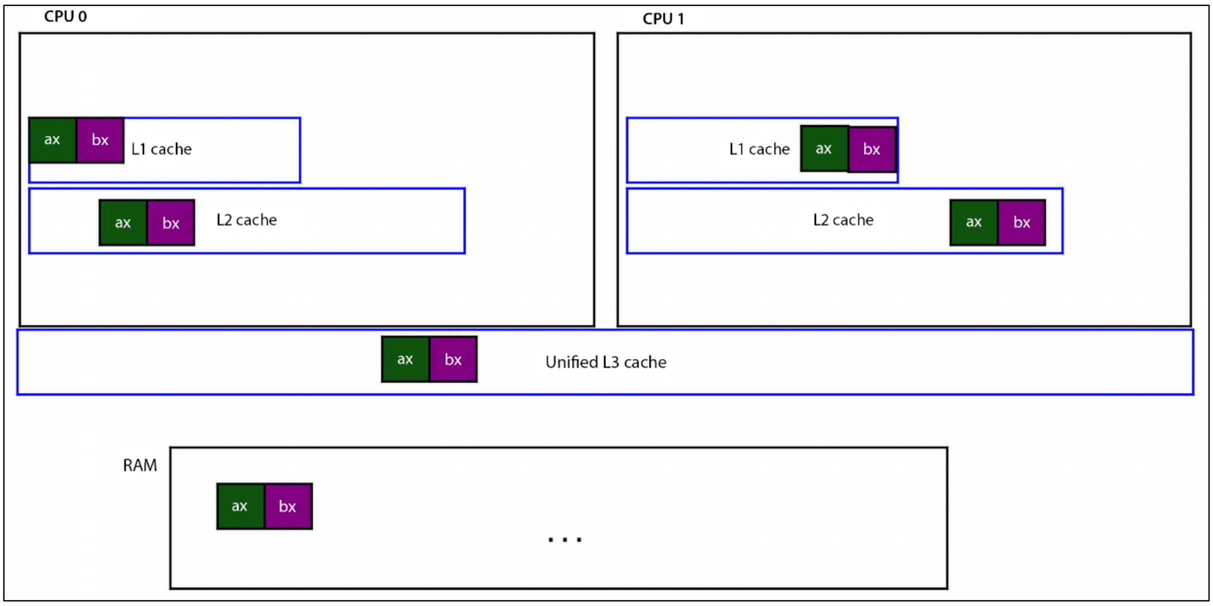
```
$ echo abc > /dev/llkd_miscdrv_rdwr_refcount ; sudo dmesg
[  137.143144] miscdrv_rdwr_refcount:miscdrv_init_refcount(): LLKD misc driver (major # 10) registered, minor# = 120,
 dev node is llkd_miscdrv_rdwr_refcount
[  137.143149] misc llkd_miscdrv_rdwr_refcount: A sample print via the dev_dbg(): driver initialized
[  142.155554] miscdrv_rdwr_refcount:open_miscdrv_rdwr(): 002)  bash :1474   |  ...0   /* open_miscdrv_rdwr() */
[  142.155559] miscdrv_rdwr_refcount:open_miscdrv_rdwr(): *** Bad case! About to overflow refcount var! ***
[  142.155560] ------------[ cut here ]------------
[  142.155561] refcount_t: saturated; leaking memory.
[  142.155567] WARNING: CPU: 2 PID: 1474 at lib/refcount.c:22 refcount_warn_saturate+0x148/0x150
[  142.155572] Modules linked in: miscdrv_rdwr_refcount(OE) binfmt_misc nls_iso8859_1 snd_intel8x0 snd_ac97_codec ac9
7_bus snd_pcm snd_seq snd_seq_device intel_rapl_msr snd_timer intel_rapl_common crct10dif_pclmul crc32_pclmul polyval
_clmulni polyval_generic snd ghash_clmulni_intel aesni_intel crypto_simd cryptd rapl video wmi i2c_piix4 soundcore vb
oxguest(OE) joydev input_leds mac_hid serio_raw vmwgfx drm_kms_helper syscopyarea sysfillrect sysimgblt fb_sys_fops d
rm_ttm_helper ttm drm msr parport_pc ppdev lp parport efi_pstore dmi_sysfs ip_tables x_tables autofs4 hid_generic usb
hid hid psmouse ahci e1000 libahci pata_acpi
[  142.155605] CPU: 2 PID: 1474 Comm: bash Tainted: G           OE      6.1.25-dbg #2
[  142.155607] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[  142.155608] RIP: 0010:refcount_warn_saturate+0x148/0x150
[  142.155610] Code: b8 77 01 8d c6 05 5e 36 6c 01 01 e8 e2 f2 9b ff 0f 0b e9 38 ff ff ff 48 c7 c7 90 77 01 8d c6 05
45 36 6c 01 01 e8 c8 f2 9b ff <0f> 0b e9 1e ff ff ff 90 8b 07 3d 00 00 00 c0 74 12 83 f8 01 74 1d
[  142.155611] RSP: 0018:ffffb7a14291baf0 EFLAGS: 00010246
[  142.155613] RAX: 0000000000000000 RBX: 0000000000000000 RCX: 0000000000000000
[  142.155614] RDX: 0000000000000000 RSI: 0000000000000000 RDI: 0000000000000000
[  142.155615] RBP: ffffb7a14291baf8 R08: 0000000000000000 R09: 0000000000000000
[  142.155616] R10: 0000000000000000 R11: 0000000000000000 R12: 0000000000000000
[  142.155617] R13: ffff96c0c36c9000 R14: ffff96c0d95e5028 R15: ffffffffc0671700
[  142.155618] FS:  00007f1129306740(0000) GS:ffff96c13dc80000(0000) knlGS:0000000000000000
[  142.155619] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[  142.155620] CR2: 0000558aa6ccde24 CR3: 000000000e2ea001 CR4: 00000000000706e0
[  142.155623] Call Trace:
[  142.155624]  <TASK>
[  142.155627]  open_miscdrv_rdwr+0x153/0x1d0 [miscdrv_rdwr_refcount]
[  142.155631]  misc_open+0x127/0x150
```

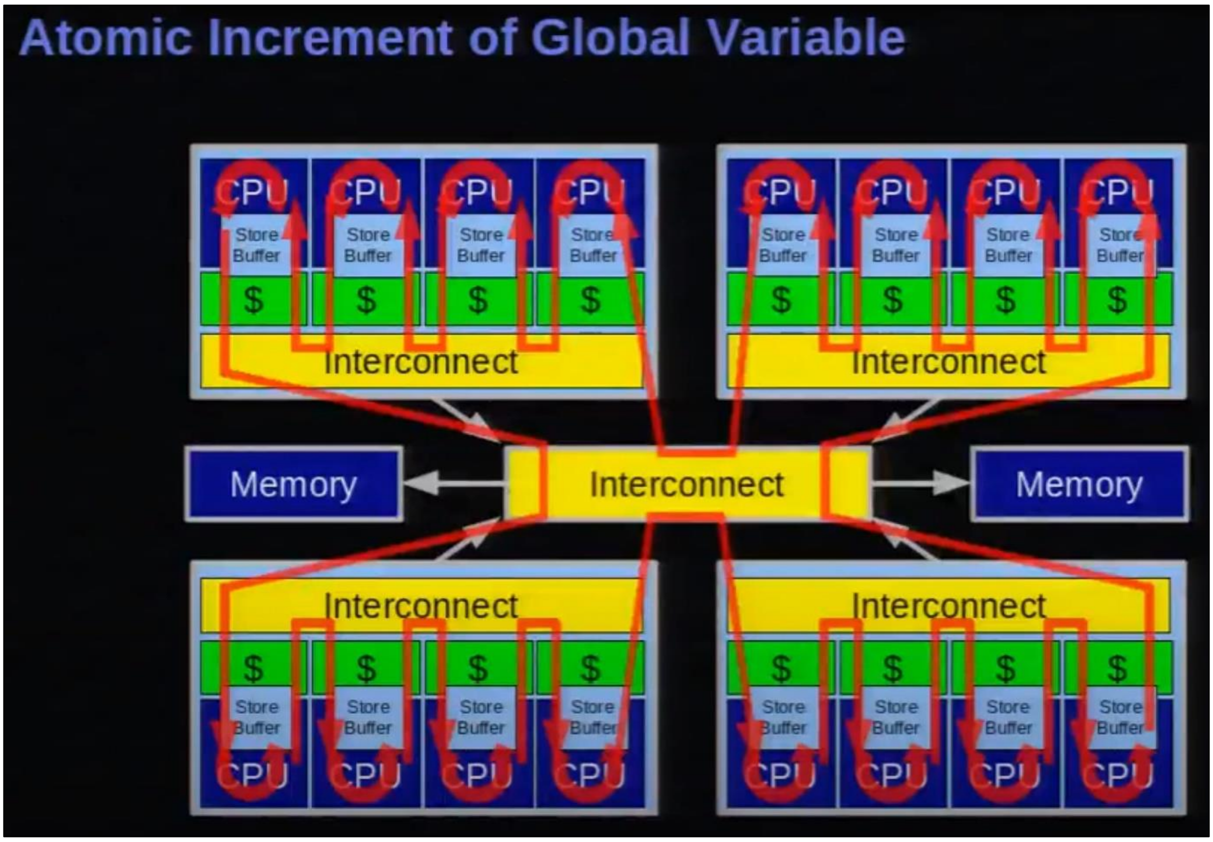


```
1_rmw_atomic_bitops: inserted
 1:                    at init: mem :   0 = 0x00
 2:optimal: via set_bit(7,&mem): mem : 128 = 0x80
delta: 29 ns
 3: set msb suboptimal: 7,&mem: mem : 128 = 0x80
delta: 125 ns
 4:          clear_bit(7,&mem): mem :   0 = 0x00
 5:         change_bit(7,&mem): mem : 128 = 0x80
 6:   test_and_set_bit(0,&mem): mem : 129 = 0x81
       ret = 0
 7: test_and_clear_bit(0,&mem): mem : 128 = 0x80
       ret (prev value of bit 0) = 1
 8:test_and_change_bit(1,&mem): mem : 130 = 0x82
       ret (prev value of bit 1) = 0
 9: test_bit(7-0,&mem):
  bit 7 (0x80) : set
  bit 6 (0x40) : cleared
  bit 5 (0x20) : cleared
  bit 4 (0x10) : cleared
  bit 3 (0x08) : cleared
  bit 2 (0x04) : cleared
  bit 1 (0x02) : set
  bit 0 (0x01) : cleared
```

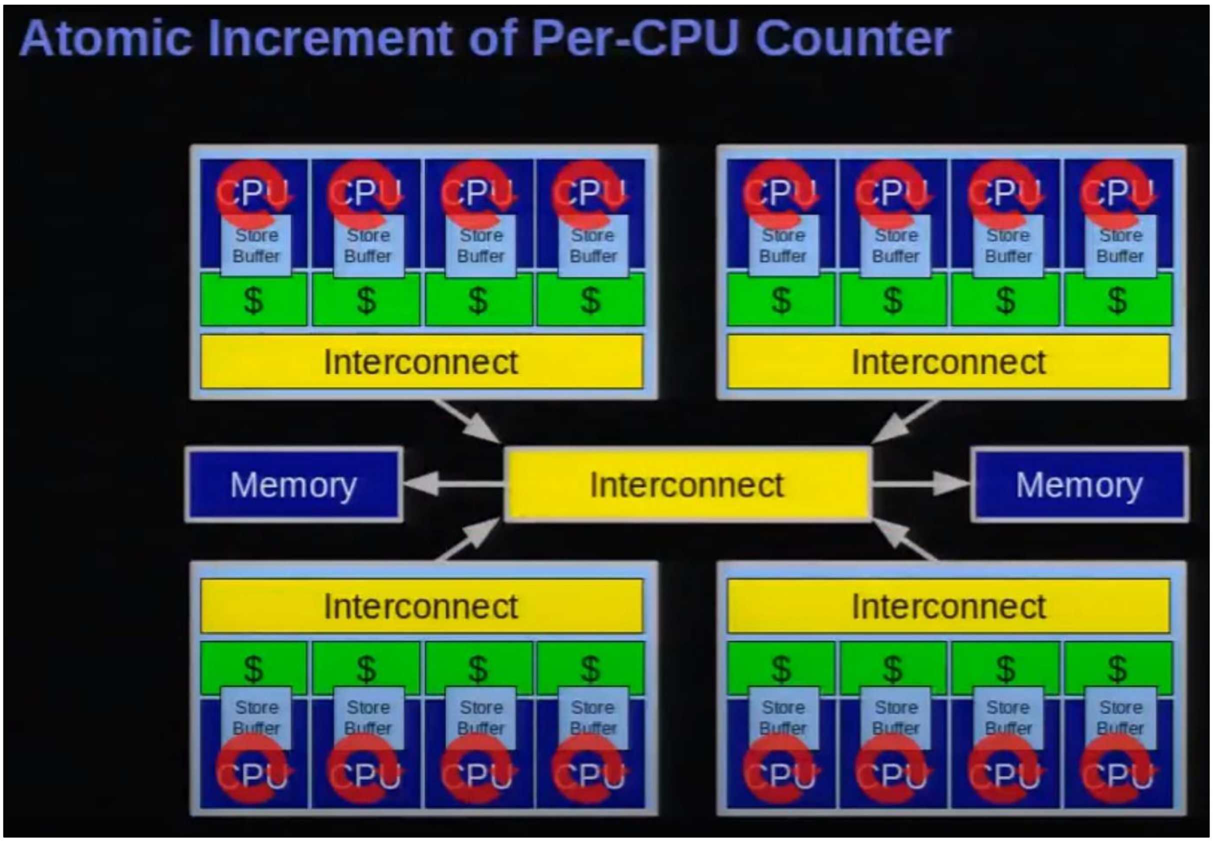

L1
CPU core 0
L1
CPU core 1
L2
L2
L3
RAM


L1
CPU core 0
L1
0-63
CPU core 1
L2
L2
0-63
L3
0-63
char
0-63
myarr[264], x;
RAM



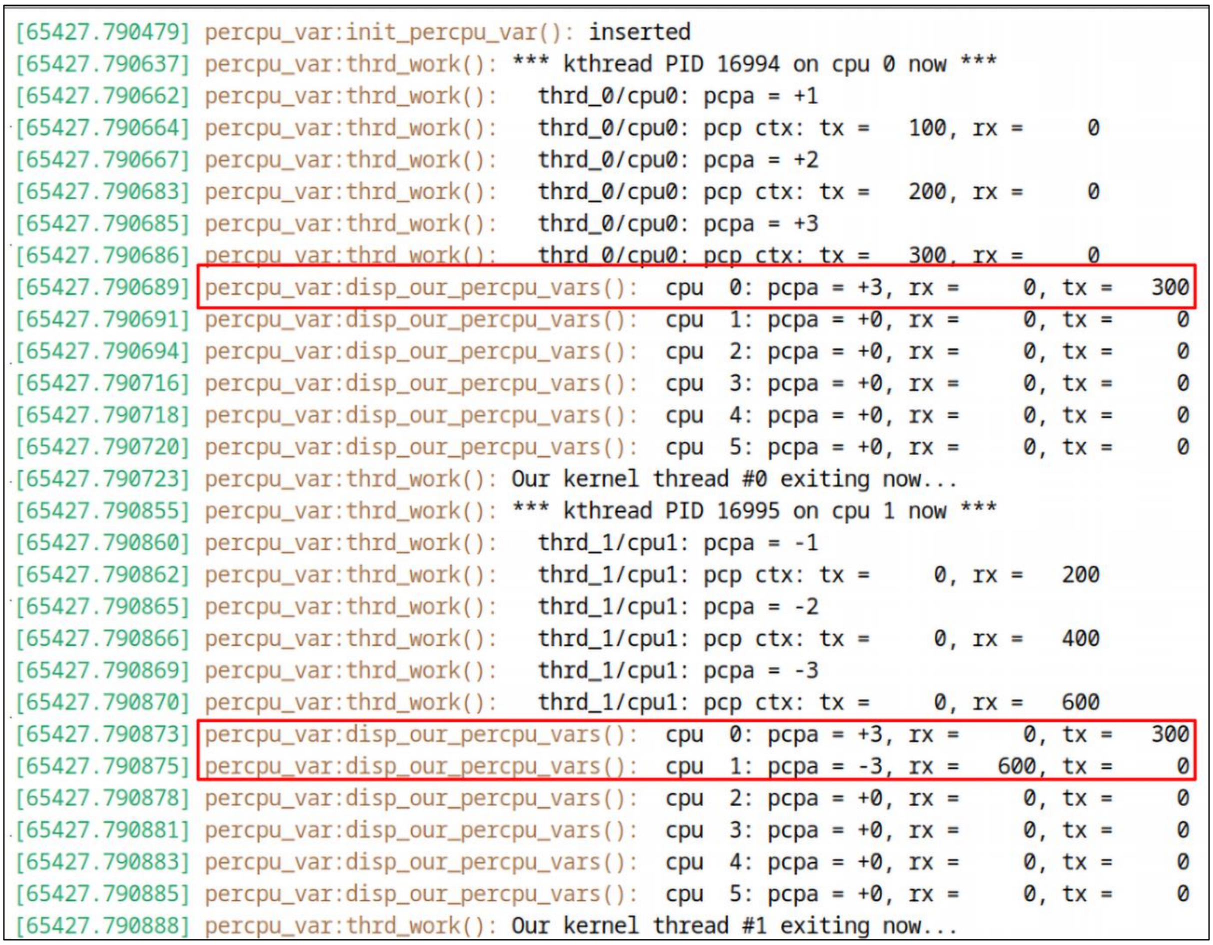


Atomic Increment of Global Variable

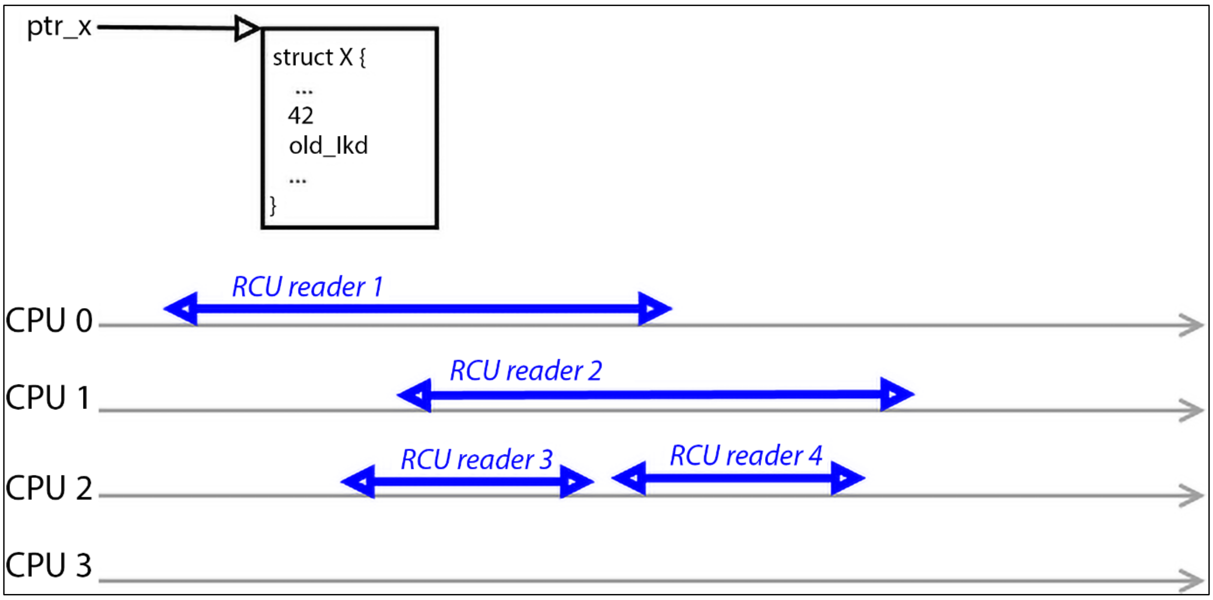


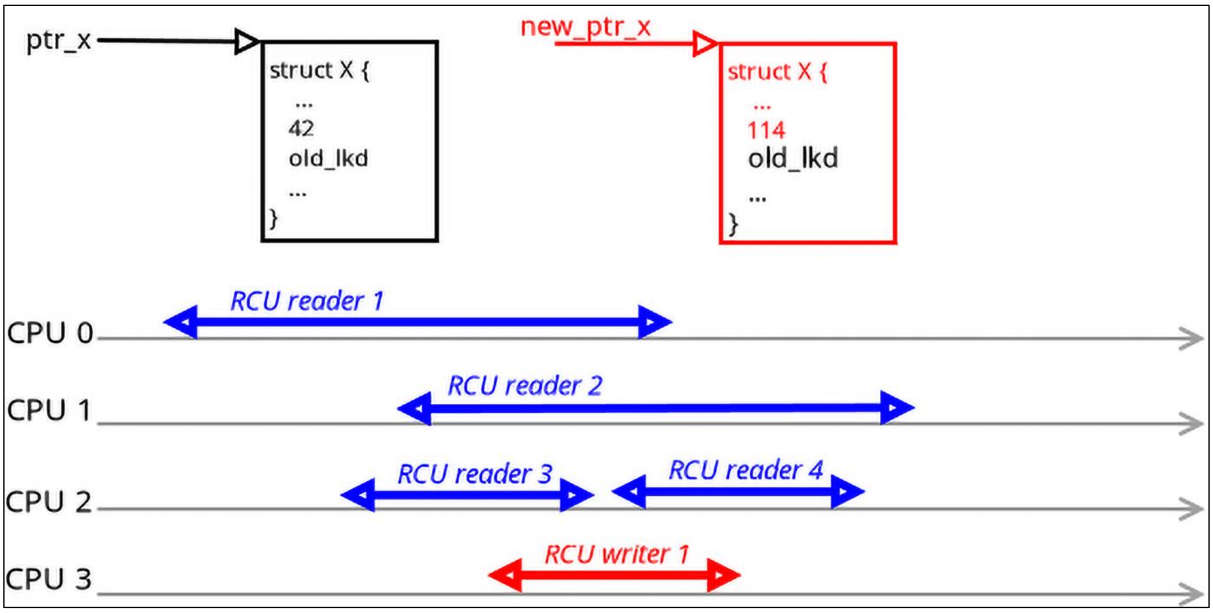
Atomic Increment of Per-CPU Counter

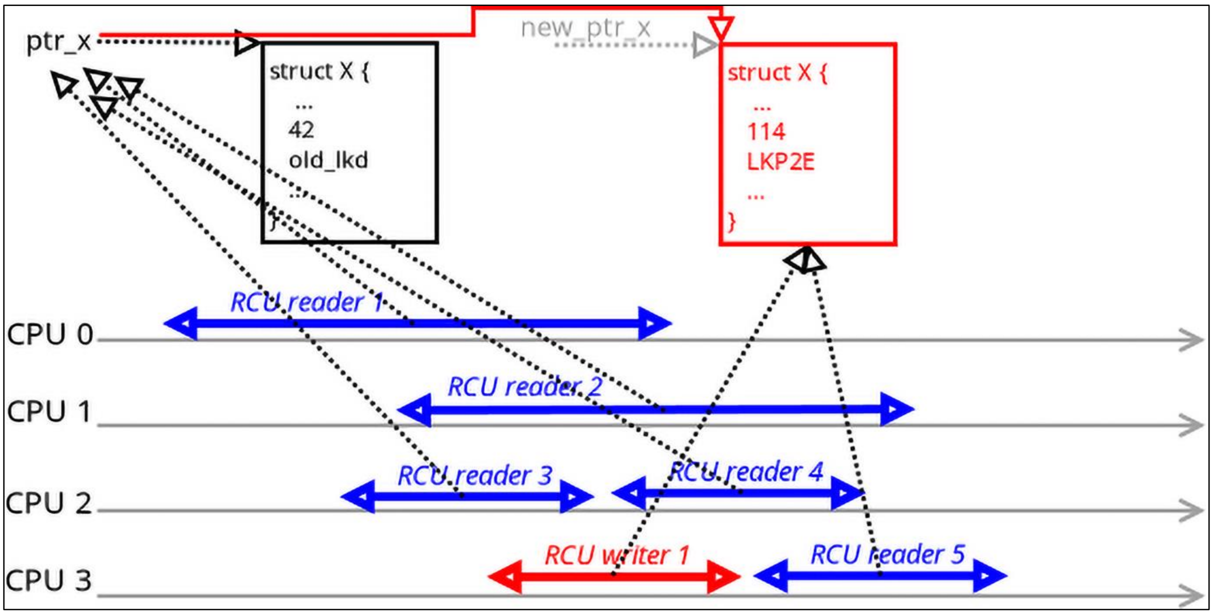
pcpa=0	pcpa=0	pcpa=0	pcpa=0
CPU 0	CPU 1	CPU 2	CPU 3

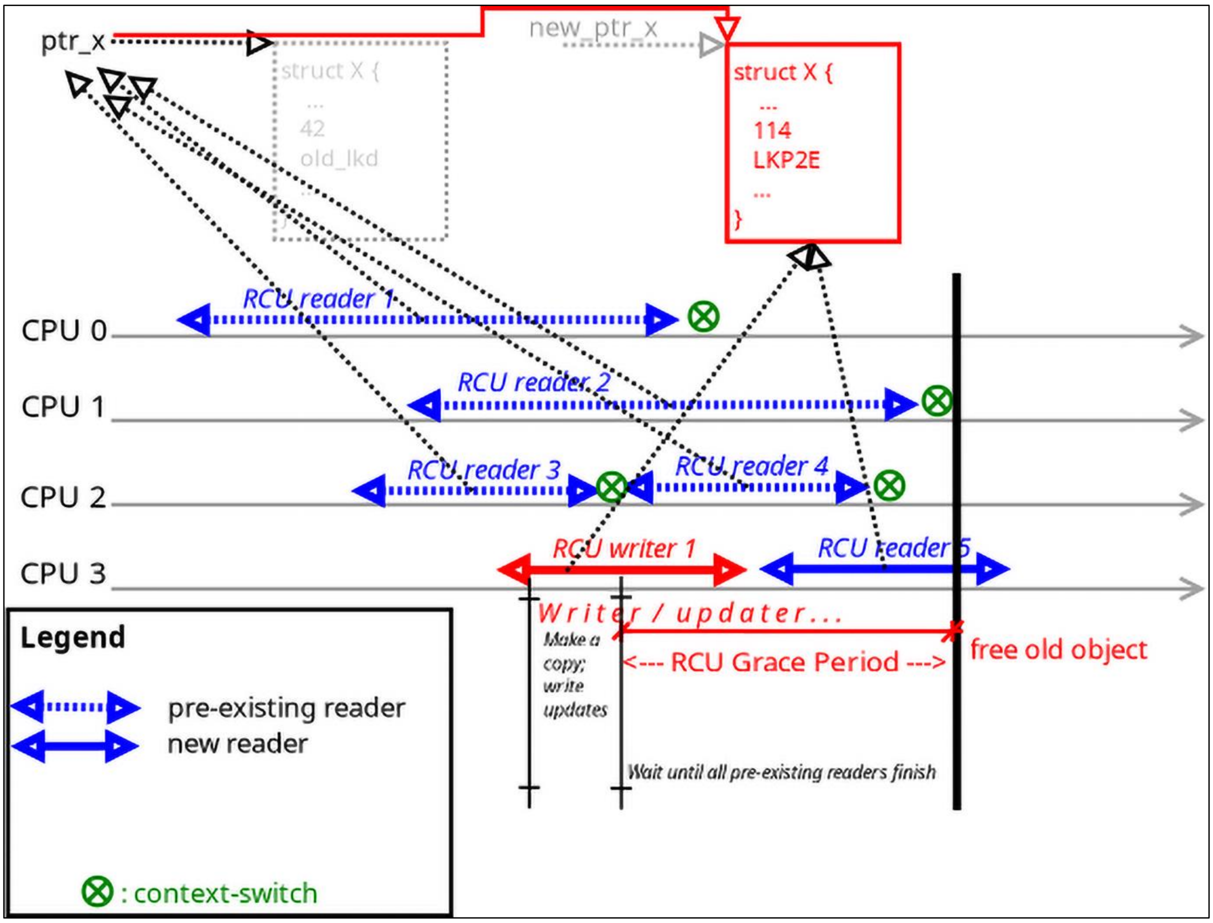


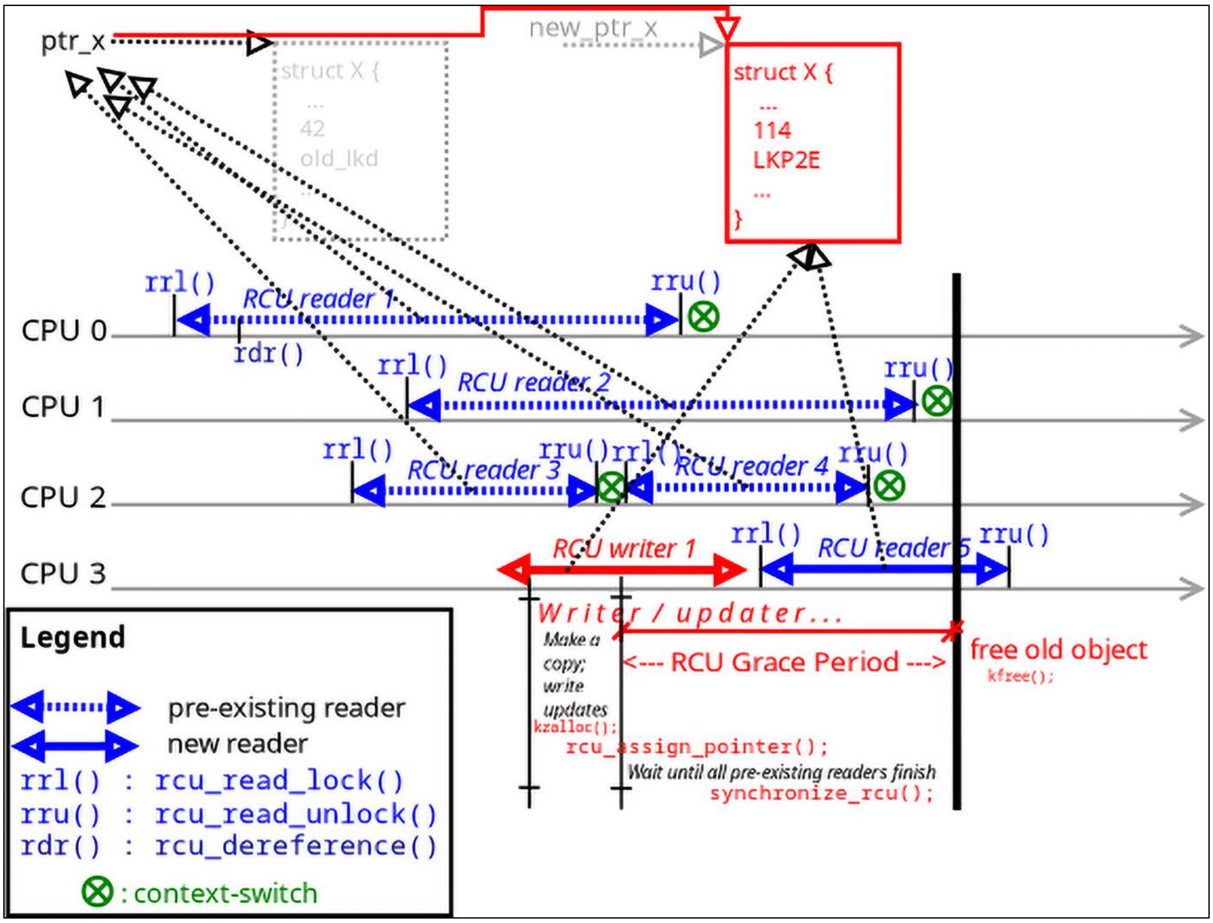
```
[65427.790479] percpu_var:init_percpu_var(): inserted
[65427.790637] percpu_var:thrd_work(): *** kthread PID 16994 on cpu 0 now ***
[65427.790662] percpu_var:thrd_work():   thrd_0/cpu0: pcpa = +1
[65427.790664] percpu_var:thrd_work():   thrd_0/cpu0: pcp ctx: tx =   100, rx =     0
[65427.790667] percpu_var:thrd_work():   thrd_0/cpu0: pcpa = +2
[65427.790683] percpu_var:thrd_work():   thrd_0/cpu0: pcp ctx: tx =   200, rx =     0
[65427.790685] percpu_var:thrd_work():   thrd_0/cpu0: pcpa = +3
[65427.790686] percpu_var:thrd_work():   thrd_0/cpu0: pcp ctx: tx =   300, rx =     0
[65427.790689] percpu_var:disp_our_percpu_vars():  cpu  0: pcpa = +3, rx =      0, tx =   300
[65427.790691] percpu_var:disp_our_percpu_vars():  cpu  1: pcpa = +0, rx =      0, tx =     0
[65427.790694] percpu_var:disp_our_percpu_vars():  cpu  2: pcpa = +0, rx =      0, tx =     0
[65427.790716] percpu_var:disp_our_percpu_vars():  cpu  3: pcpa = +0, rx =      0, tx =     0
[65427.790718] percpu_var:disp_our_percpu_vars():  cpu  4: pcpa = +0, rx =      0, tx =     0
[65427.790720] percpu_var:disp_our_percpu_vars():  cpu  5: pcpa = +0, rx =      0, tx =     0
[65427.790723] percpu_var:thrd_work(): Our kernel thread #0 exiting now...
[65427.790855] percpu_var:thrd_work(): *** kthread PID 16995 on cpu 1 now ***
[65427.790860] percpu_var:thrd_work():   thrd_1/cpu1: pcpa = -1
[65427.790862] percpu_var:thrd_work():   thrd_1/cpu1: pcp ctx: tx =     0, rx =   200
[65427.790865] percpu_var:thrd_work():   thrd_1/cpu1: pcpa = -2
[65427.790866] percpu_var:thrd_work():   thrd_1/cpu1: pcp ctx: tx =     0, rx =   400
[65427.790869] percpu_var:thrd_work():   thrd_1/cpu1: pcpa = -3
[65427.790870] percpu_var:thrd_work():   thrd_1/cpu1: pcp ctx: tx =     0, rx =   600
[65427.790873] percpu_var:disp_our_percpu_vars():  cpu  0: pcpa = +3, rx =      0, tx =   300
[65427.790875] percpu_var:disp_our_percpu_vars():  cpu  1: pcpa = -3, rx =    600, tx =     0
[65427.790878] percpu_var:disp_our_percpu_vars():  cpu  2: pcpa = +0, rx =      0, tx =     0
[65427.790881] percpu_var:disp_our_percpu_vars():  cpu  3: pcpa = +0, rx =      0, tx =     0
[65427.790883] percpu_var:disp_our_percpu_vars():  cpu  4: pcpa = +0, rx =      0, tx =     0
[65427.790885] percpu_var:disp_our_percpu_vars():  cpu  5: pcpa = +0, rx =      0, tx =     0
[65427.790888] percpu_var:thrd_work(): Our kernel thread #1 exiting now...
```

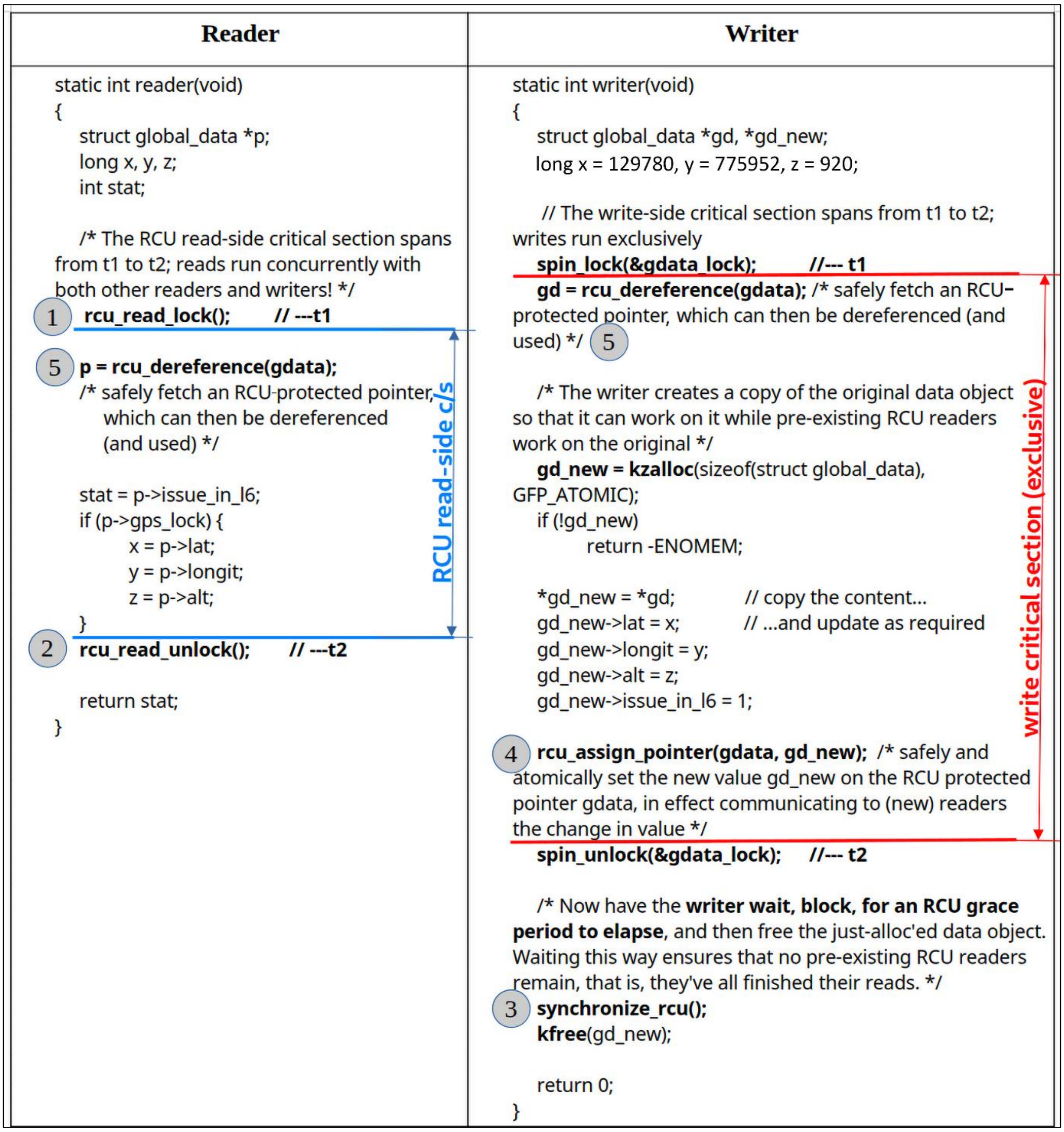



ptr_x
new_ptr_x
struct X {
...
42
old_lkd
...
}
struct X {
...
114
old_lkd
...
}
RCU reader 1
CPU 0
RCU reader 2
CPU 1
RCU reader 3
RCU reader 4
CPU 2
RCU writer 1
CPU 3


ptr_x
new_ptr_x
struct X {
...
42
old_lkd
...
}
struct X {
...
114
LKP2E
...
}
RCU reader 1
CPU 0
RCU reader 2
CPU 1
RCU reader 3
RCU reader 4
CPU 2
RCU writer 1
RCU reader 5
CPU 3


ptr_x
new_ptr_x
struct X {
...
42
old_lkd
...
}
struct X {
...
114
LKP2E
...
}
CPU 0
CPU 1
CPU 2
CPU 3
RCU reader 1
RCU reader 2
RCU reader 3
RCU reader 4
RCU writer 1
RCU reader 5
Writer / updater...
Make a copy; write updates
<--- RCU Grace Period --->
free old object
Wait until all pre-existing readers finish
Legend
pre-existing reader
new reader
⊗ : context-switch


ptr_x
new_ptr_x
struct X {
...
42
old_lkd
...
}
struct X {
...
114
LKP2E
...
}
rrl()
RCU reader 1
rru()
rdr()
CPU 0
rrl()
RCU reader 2
rru()
CPU 1
rrl()
RCU reader 3
rru()
rrl()
RCU reader 4
rru()
CPU 2
RCU writer 1
rrl()
RCU reader 5
rru()
CPU 3
Writer / updater...
Make a copy; write updates
kzalloc();
rcu_assign_pointer();
<--- RCU Grace Period --->
free old object
kfree();
Wait until all pre-existing readers finish
synchronize_rcu();
Legend
pre-existing reader
new reader
rrl() : rcu_read_lock()
rru() : rcu_read_unlock()
rdr() : rcu_dereference()
⊗ : context-switch

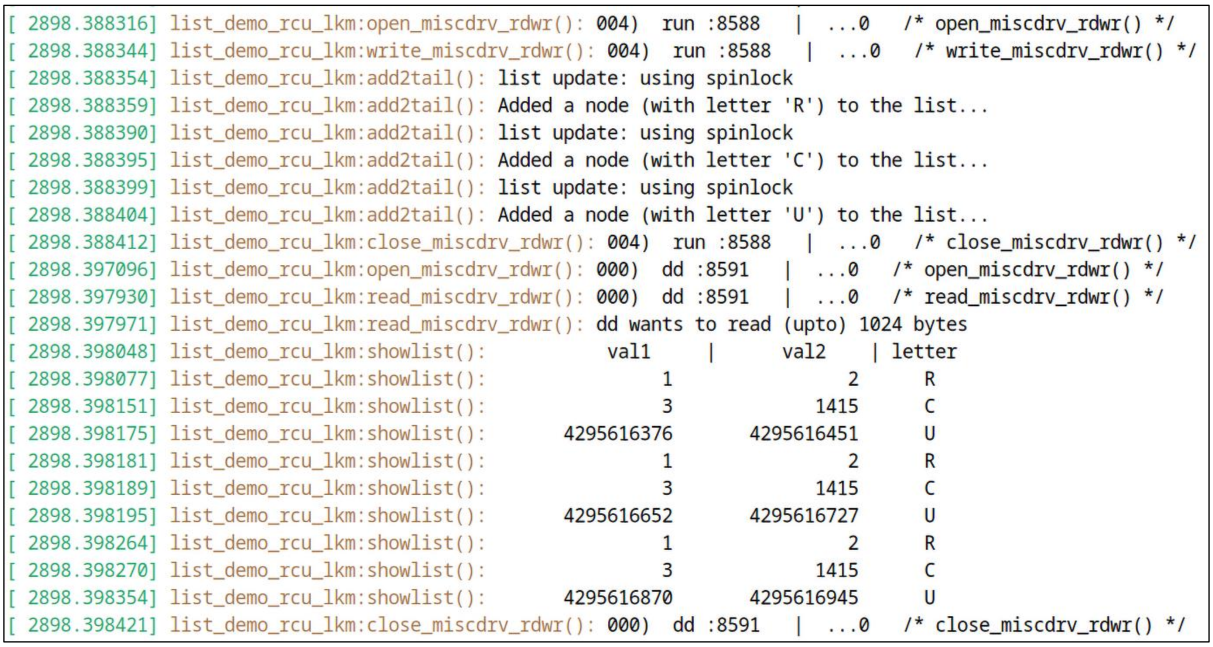


Reader

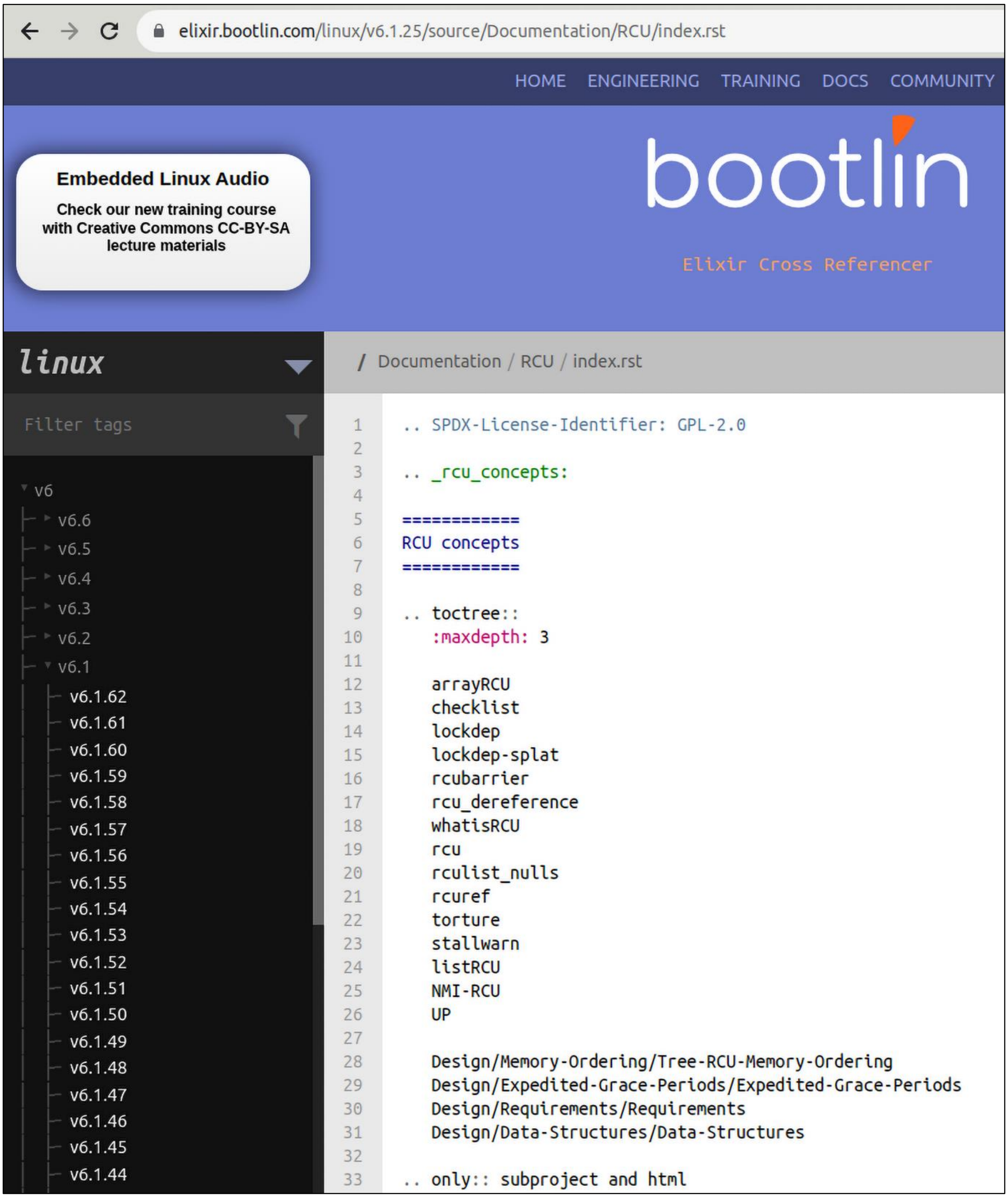
```
static int reader(void)
{
    struct global_data *p;
    long x, y, z;
    int stat;

    /* The RCU read-side critical section spans
from t1 to t2; reads run concurrently with
both other readers and writers! */
    rcu_read_lock();       // ---t1

    p = rcu_dereference(gdata);
    /* safely fetch an RCU-protected pointer,
       which can then be dereferenced
       (and used) */

    stat = p->issue_in_l6;
    if (p->gps_lock) {
            x = p->lat;
            y = p->longit;
            z = p->alt;
    }
    rcu_read_unlock();       // ---t2

    return stat;
}
```

Writer

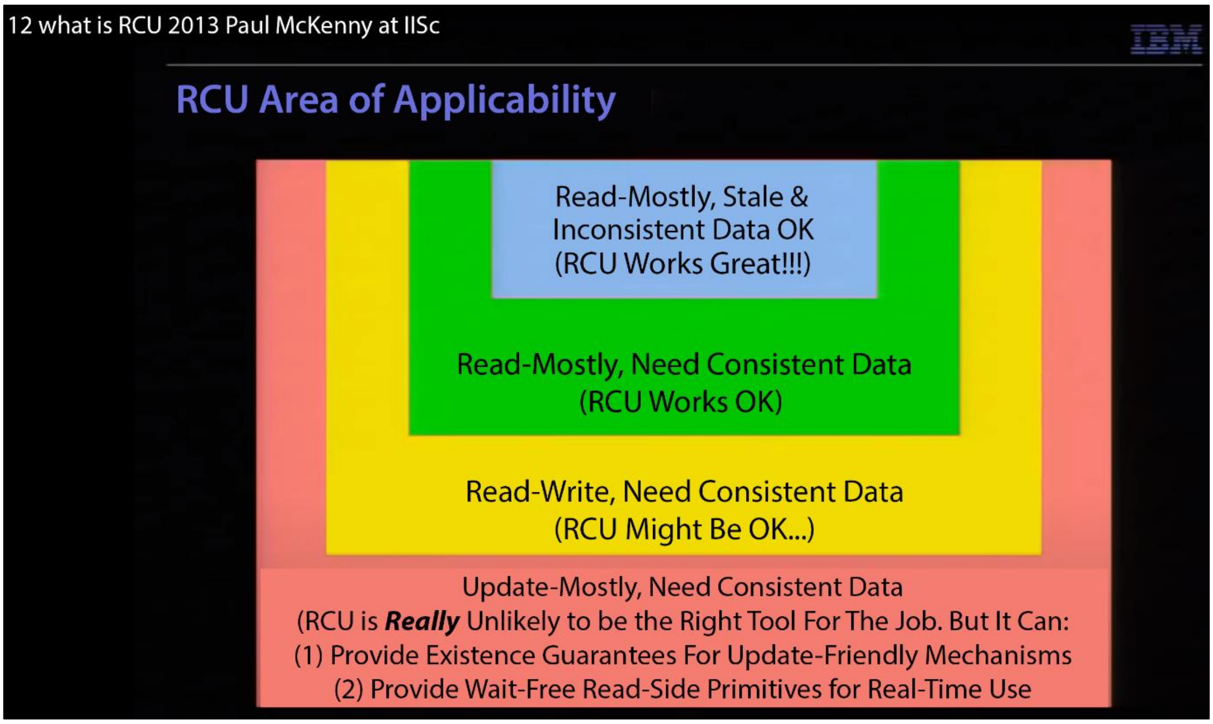
```
static int writer(void)
{
    struct global_data *gd, *gd_new;
    long x = 129780, y = 775952, z = 920;

     // The write-side critical section spans from t1 to t2;
writes run exclusively
    spin_lock(&gdata_lock);       //--- t1
    gd = rcu_dereference(gdata); /* safely fetch an RCU-
protected pointer, which can then be dereferenced (and
used) */

    /* The writer creates a copy of the original data object
so that it can work on it while pre-existing RCU readers
work on the original */
    gd_new = kzalloc(sizeof(struct global_data),
GFP_ATOMIC);
    if (!gd_new)
            return -ENOMEM;

    *gd_new = *gd;          // copy the content...
    gd_new->lat = x;        // ...and update as required
    gd_new->longit = y;
    gd_new->alt = z;
    gd_new->issue_in_l6 = 1;

    rcu_assign_pointer(gdata, gd_new);  /* safely and
atomically set the new value gd_new on the RCU protected
pointer gdata, in effect communicating to (new) readers
the change in value */
    spin_unlock(&gdata_lock);     //--- t2

    /* Now have the writer wait, block, for an RCU grace
period to elapse, and then free the just-alloc'ed data object.
Waiting this way ensures that no pre-existing RCU readers
remain, that is, they've all finished their reads. */
    synchronize_rcu();
    kfree(gd_new);

    return 0;
}
```



```
[ 2898.388316] list_demo_rcu_lkm:open_miscdrv_rdwr(): 004)  run :8588   |  ...0   /* open_miscdrv_rdwr() */
[ 2898.388344] list_demo_rcu_lkm:write_miscdrv_rdwr(): 004)  run :8588   |  ...0   /* write_miscdrv_rdwr() */
[ 2898.388354] list_demo_rcu_lkm:add2tail(): list update: using spinlock
[ 2898.388359] list_demo_rcu_lkm:add2tail(): Added a node (with letter 'R') to the list...
[ 2898.388390] list_demo_rcu_lkm:add2tail(): list update: using spinlock
[ 2898.388395] list_demo_rcu_lkm:add2tail(): Added a node (with letter 'C') to the list...
[ 2898.388399] list_demo_rcu_lkm:add2tail(): list update: using spinlock
[ 2898.388404] list_demo_rcu_lkm:add2tail(): Added a node (with letter 'U') to the list...
[ 2898.388412] list_demo_rcu_lkm:close_miscdrv_rdwr(): 004)  run :8588   |  ...0   /* close_miscdrv_rdwr() */
[ 2898.397096] list_demo_rcu_lkm:open_miscdrv_rdwr(): 000)  dd :8591   |  ...0   /* open_miscdrv_rdwr() */
[ 2898.397930] list_demo_rcu_lkm:read_miscdrv_rdwr(): 000)  dd :8591   |  ...0   /* read_miscdrv_rdwr() */
[ 2898.397971] list_demo_rcu_lkm:read_miscdrv_rdwr(): dd wants to read (upto) 1024 bytes
[ 2898.398048] list_demo_rcu_lkm:showlist():          val1     |       val2     | letter
[ 2898.398077] list_demo_rcu_lkm:showlist():             1                  2       R
[ 2898.398151] list_demo_rcu_lkm:showlist():             3               1415       C
[ 2898.398175] list_demo_rcu_lkm:showlist():    4295616376         4295616451       U
[ 2898.398181] list_demo_rcu_lkm:showlist():             1                  2       R
[ 2898.398189] list_demo_rcu_lkm:showlist():             3               1415       C
[ 2898.398195] list_demo_rcu_lkm:showlist():    4295616652         4295616727       U
[ 2898.398264] list_demo_rcu_lkm:showlist():             1                  2       R
[ 2898.398270] list_demo_rcu_lkm:showlist():             3               1415       C
[ 2898.398354] list_demo_rcu_lkm:showlist():    4295616870         4295616945       U
[ 2898.398421] list_demo_rcu_lkm:close_miscdrv_rdwr(): 000)  dd :8591   |  ...0   /* close_miscdrv_rdwr() */
```



```
.. SPDX-License-Identifier: GPL-2.0

.. _rcu_concepts:

============
RCU concepts
============

.. toctree::
   :maxdepth: 3

   arrayRCU
   checklist
   lockdep
   lockdep-splat
   rcubarrier
   rcu_dereference
   whatisRCU
   rcu
   rculist_nulls
   rcuref
   torture
   stallwarn
   listRCU
   NMI-RCU
   UP

   Design/Memory-Ordering/Tree-RCU-Memory-Ordering
   Design/Expedited-Grace-Periods/Expedited-Grace-Periods
   Design/Requirements/Requirements
   Design/Data-Structures/Data-Structures

.. only:: subproject and html
```



12 what is RCU 2013 Paul McKenny at IISc

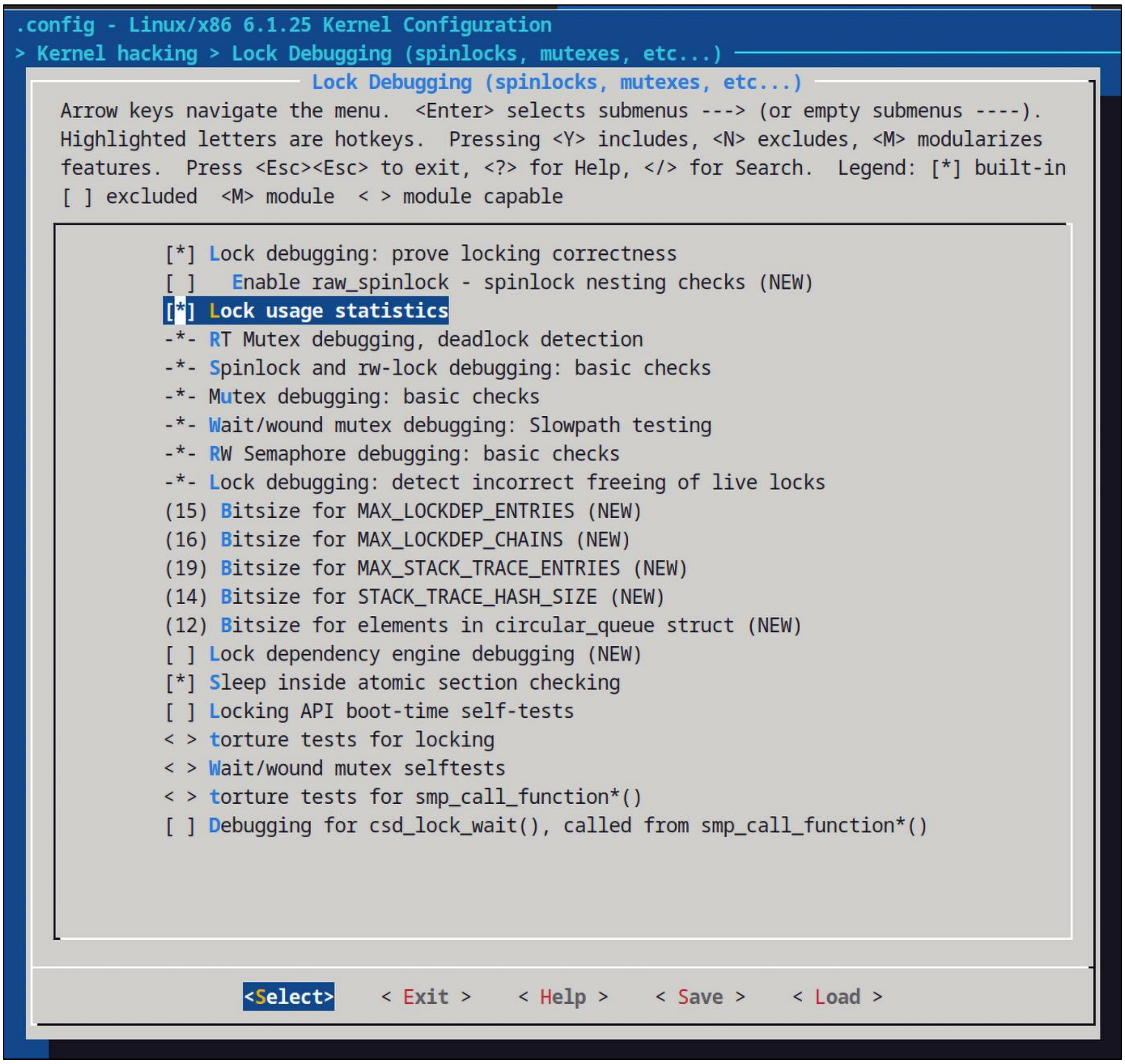
RCU Area of Applicability

Read-Mostly, Stale &
Inconsistent Data OK
(RCU Works Great!!!)

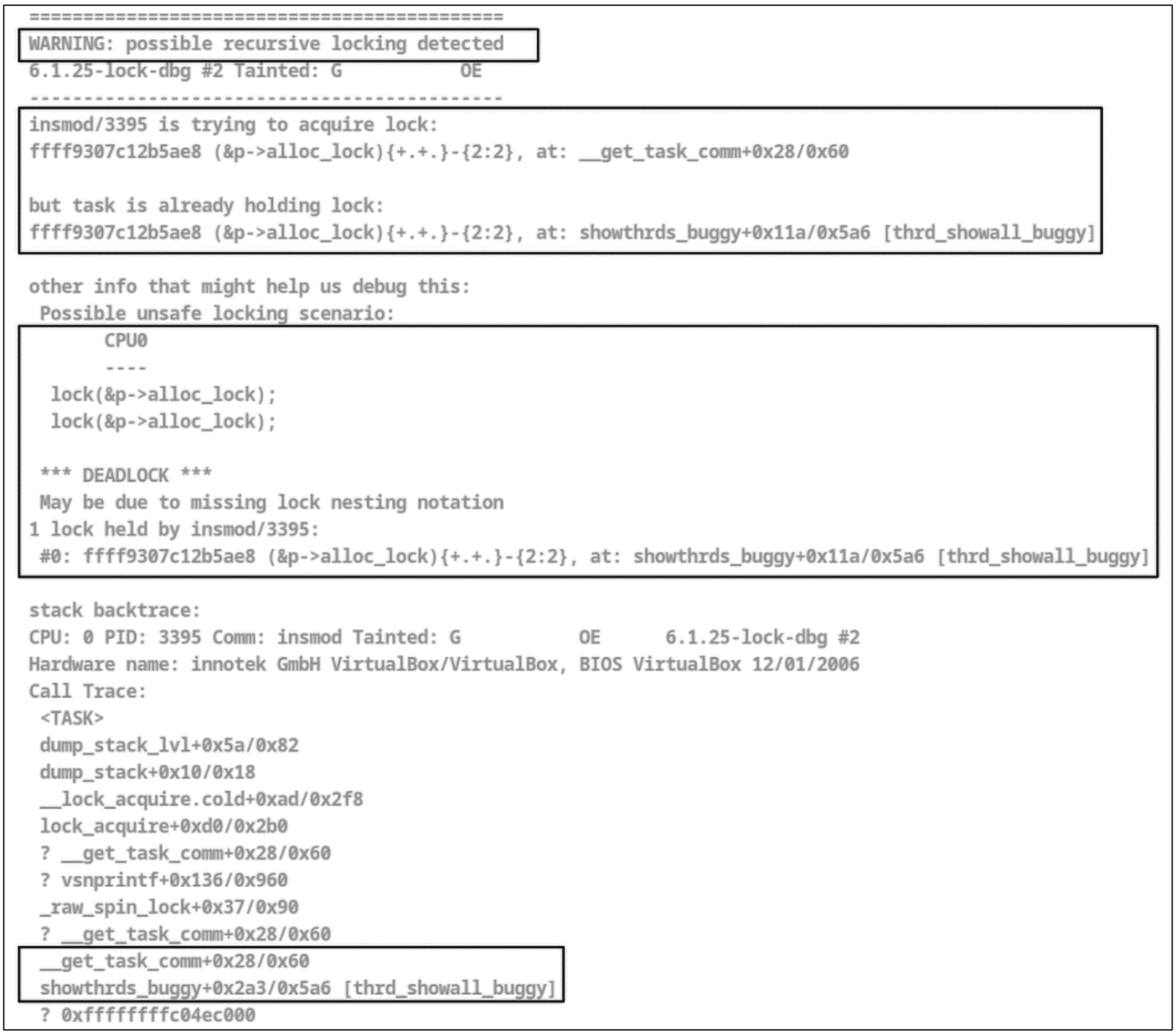
Read-Mostly, Need Consistent Data
(RCU Works OK)

Read-Write, Need Consistent Data
(RCU Might Be OK...)

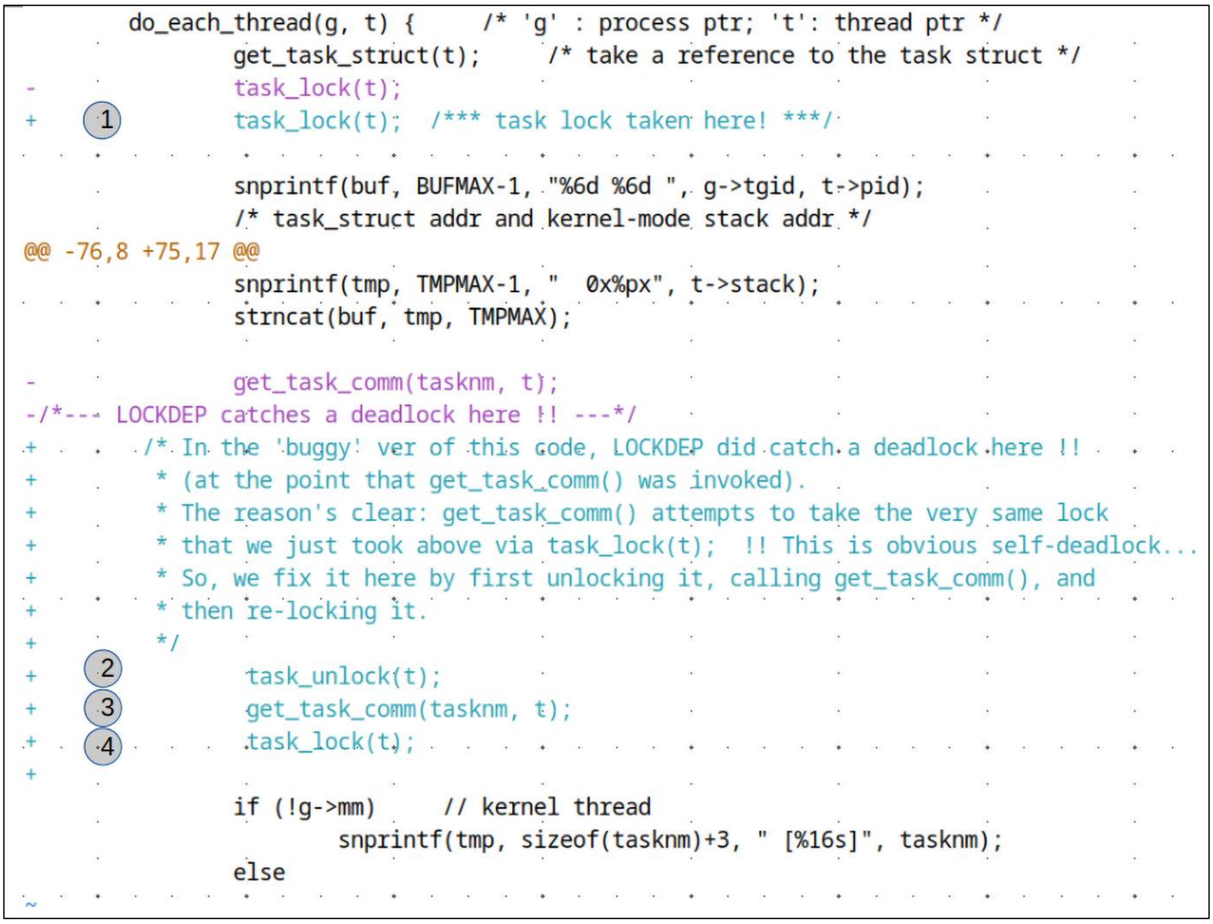
Update-Mostly, Need Consistent Data
(RCU is ***Really*** Unlikely to be the Right Tool For The Job. But It Can:
(1) Provide Existence Guarantees For Update-Friendly Mechanisms
(2) Provide Wait-Free Read-Side Primitives for Real-Time Use



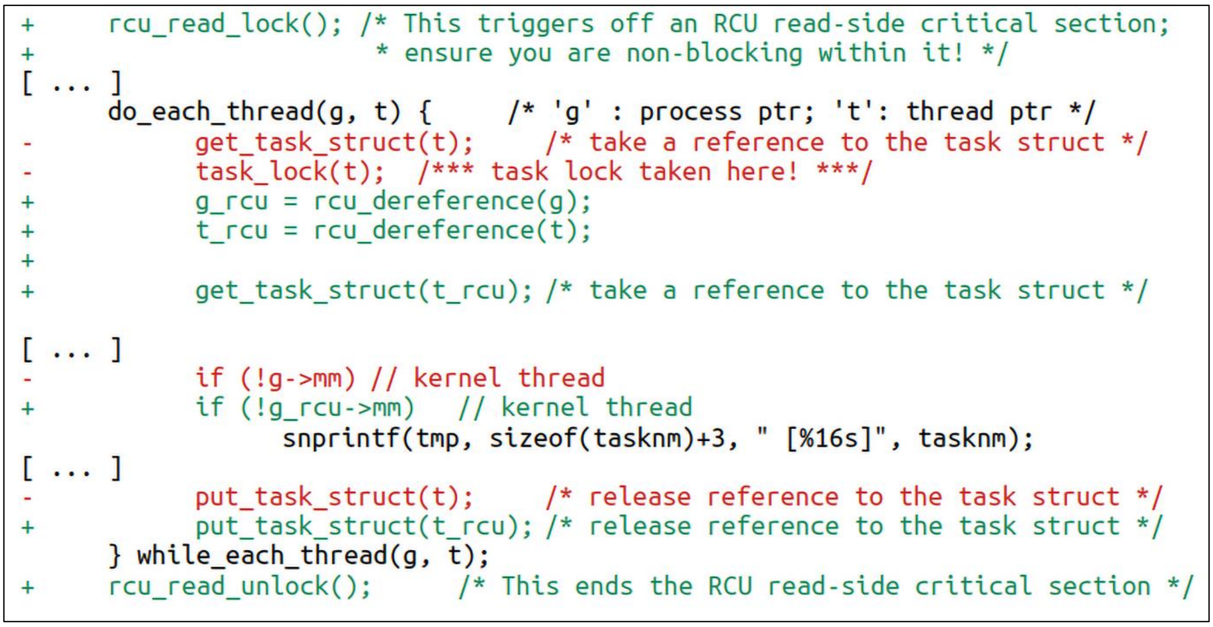
```
.config - Linux/x86 6.1.25 Kernel Configuration
> Kernel hacking > Lock Debugging (spinlocks, mutexes, etc...)
                Lock Debugging (spinlocks, mutexes, etc...)
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes
  features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in
  [ ] excluded  <M> module  < > module capable

          [*] Lock debugging: prove locking correctness
          [ ]   Enable raw_spinlock - spinlock nesting checks (NEW)
          [*] Lock usage statistics
          -*- RT Mutex debugging, deadlock detection
          -*- Spinlock and rw-lock debugging: basic checks
          -*- Mutex debugging: basic checks
          -*- Wait/wound mutex debugging: Slowpath testing
          -*- RW Semaphore debugging: basic checks
          -*- Lock debugging: detect incorrect freeing of live locks
          (15) Bitsize for MAX_LOCKDEP_ENTRIES (NEW)
          (16) Bitsize for MAX_LOCKDEP_CHAINS (NEW)
          (19) Bitsize for MAX_STACK_TRACE_ENTRIES (NEW)
          (14) Bitsize for STACK_TRACE_HASH_SIZE (NEW)
          (12) Bitsize for elements in circular_queue struct (NEW)
          [ ] Lock dependency engine debugging (NEW)
          [*] Sleep inside atomic section checking
          [ ] Locking API boot-time self-tests
          < > torture tests for locking
          < > Wait/wound mutex selftests
          < > torture tests for smp_call_function*()
          [ ] Debugging for csd_lock_wait(), called from smp_call_function*()

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



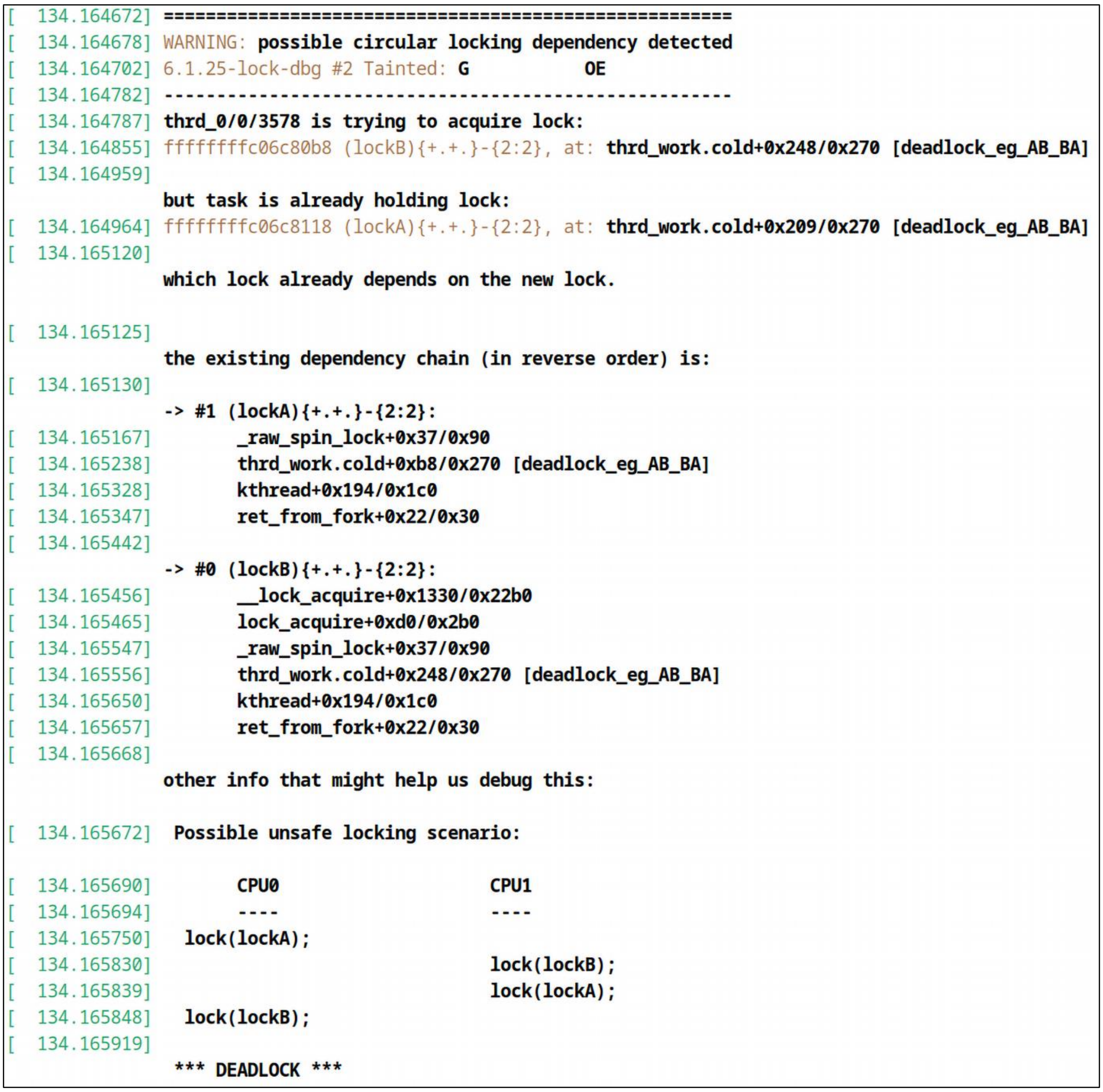
```
============================================
WARNING: possible recursive locking detected
6.1.25-lock-dbg #2 Tainted: G           OE
--------------------------------------------
insmod/3395 is trying to acquire lock:
ffff9307c12b5ae8 (&p->alloc_lock){+.+.}-{2:2}, at: __get_task_comm+0x28/0x60

but task is already holding lock:
ffff9307c12b5ae8 (&p->alloc_lock){+.+.}-{2:2}, at: showthrds_buggy+0x11a/0x5a6 [thrd_showall_buggy]

other info that might help us debug this:
 Possible unsafe locking scenario:
       CPU0
       ----
  lock(&p->alloc_lock);
  lock(&p->alloc_lock);

 *** DEADLOCK ***
 May be due to missing lock nesting notation
1 lock held by insmod/3395:
 #0: ffff9307c12b5ae8 (&p->alloc_lock){+.+.}-{2:2}, at: showthrds_buggy+0x11a/0x5a6 [thrd_showall_buggy]

stack backtrace:
CPU: 0 PID: 3395 Comm: insmod Tainted: G           OE      6.1.25-lock-dbg #2
Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
Call Trace:
 <TASK>
 dump_stack_lvl+0x5a/0x82
 dump_stack+0x10/0x18
 __lock_acquire.cold+0xad/0x2f8
 lock_acquire+0xd0/0x2b0
 ? __get_task_comm+0x28/0x60
 ? vsnprintf+0x136/0x960
 _raw_spin_lock+0x37/0x90
 ? __get_task_comm+0x28/0x60
 __get_task_comm+0x28/0x60
 showthrds_buggy+0x2a3/0x5a6 [thrd_showall_buggy]
 ? 0xffffffffc04ec000
```



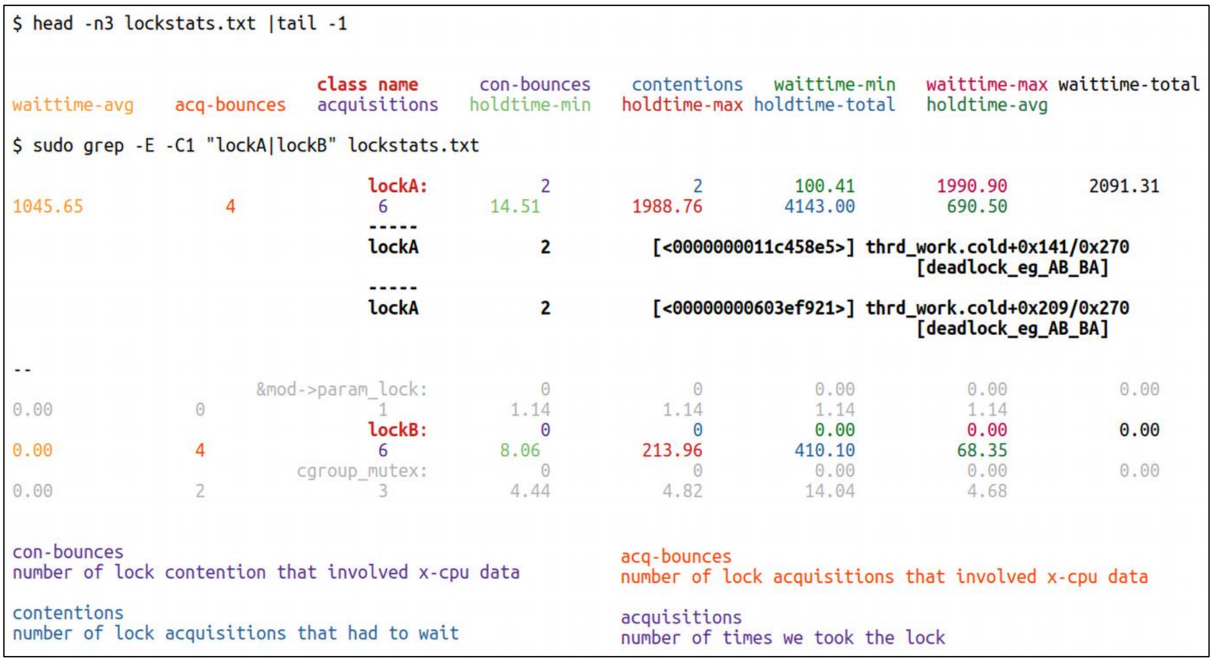
```
        do_each_thread(g, t) {     /* 'g' : process ptr; 't': thread ptr */
                get_task_struct(t);     /* take a reference to the task struct */
-               task_lock(t);
+               task_lock(t);  /*** task lock taken here! ***/

                snprintf(buf, BUFMAX-1, "%6d %6d ", g->tgid, t->pid);
                /* task_struct addr and kernel-mode stack addr */
@@ -76,8 +75,17 @@
                snprintf(tmp, TMPMAX-1, "  0x%px", t->stack);
                strncat(buf, tmp, TMPMAX);

-               get_task_comm(tasknm, t);
-/*--- LOCKDEP catches a deadlock here !! ---*/
+            /* In the 'buggy' ver of this code, LOCKDEP did catch a deadlock here !!
+             * (at the point that get_task_comm() was invoked).
+             * The reason's clear: get_task_comm() attempts to take the very same lock
+             * that we just took above via task_lock(t);  !! This is obvious self-deadlock...
+             * So, we fix it here by first unlocking it, calling get_task_comm(), and
+             * then re-locking it.
+             */
+               task_unlock(t);
+               get_task_comm(tasknm, t);
+               task_lock(t);
+
                if (!g->mm)     // kernel thread
                        snprintf(tmp, sizeof(tasknm)+3, " [%16s]", tasknm);
                else
~
```



```
+     rcu_read_lock(); /* This triggers off an RCU read-side critical section;
+                         * ensure you are non-blocking within it! */
[ ... ]
      do_each_thread(g, t) {     /* 'g' : process ptr; 't': thread ptr */
-           get_task_struct(t);     /* take a reference to the task struct */
-           task_lock(t);  /*** task lock taken here! ***/
+           g_rcu = rcu_dereference(g);
+           t_rcu = rcu_dereference(t);
+
+           get_task_struct(t_rcu); /* take a reference to the task struct */

[ ... ]
-           if (!g->mm) // kernel thread
+           if (!g_rcu->mm)   // kernel thread
                  snprintf(tmp, sizeof(tasknm)+3, " [%16s]", tasknm);
[ ... ]
-           put_task_struct(t);     /* release reference to the task struct */
+           put_task_struct(t_rcu); /* release reference to the task struct */
      } while_each_thread(g, t);
+     rcu_read_unlock();      /* This ends the RCU read-side critical section */
```



```
[  134.164672] ========================================================
[  134.164678] WARNING: possible circular locking dependency detected
[  134.164702] 6.1.25-lock-dbg #2 Tainted: G           OE
[  134.164782] --------------------------------------------------------
[  134.164787] thrd_0/0/3578 is trying to acquire lock:
[  134.164855] ffffffffc06c80b8 (lockB){+.+.}-{2:2}, at: thrd_work.cold+0x248/0x270 [deadlock_eg_AB_BA]
[  134.164959]
               but task is already holding lock:
[  134.164964] ffffffffc06c8118 (lockA){+.+.}-{2:2}, at: thrd_work.cold+0x209/0x270 [deadlock_eg_AB_BA]
[  134.165120]
               which lock already depends on the new lock.

[  134.165125]
               the existing dependency chain (in reverse order) is:
[  134.165130]
               -> #1 (lockA){+.+.}-{2:2}:
[  134.165167]        _raw_spin_lock+0x37/0x90
[  134.165238]        thrd_work.cold+0xb8/0x270 [deadlock_eg_AB_BA]
[  134.165328]        kthread+0x194/0x1c0
[  134.165347]        ret_from_fork+0x22/0x30
[  134.165442]
               -> #0 (lockB){+.+.}-{2:2}:
[  134.165456]        __lock_acquire+0x1330/0x22b0
[  134.165465]        lock_acquire+0xd0/0x2b0
[  134.165547]        _raw_spin_lock+0x37/0x90
[  134.165556]        thrd_work.cold+0x248/0x270 [deadlock_eg_AB_BA]
[  134.165650]        kthread+0x194/0x1c0
[  134.165657]        ret_from_fork+0x22/0x30
[  134.165668]
               other info that might help us debug this:

[  134.165672]  Possible unsafe locking scenario:

[  134.165690]        CPU0                    CPU1
[  134.165694]        ----                    ----
[  134.165750]   lock(lockA);
[  134.165830]                                lock(lockB);
[  134.165839]                                lock(lockA);
[  134.165848]   lock(lockB);
[  134.165919]
                *** DEADLOCK ***
```



```
$ head -n3 lockstats.txt |tail -1

                         class name      con-bounces    contentions   waittime-min    waittime-max waittime-total
waittime-avg    acq-bounces  acquisitions   holdtime-min   holdtime-max holdtime-total   holdtime-avg

$ sudo grep -E -C1 "lockA|lockB" lockstats.txt

                              lockA:               2              2         100.41         1990.90        2091.31
1045.65              4            6            14.51        1988.76        4143.00          690.50
                              -----
                              lockA            2            [<0000000011c458e5>] thrd_work.cold+0x141/0x270
                                                                                 [deadlock_eg_AB_BA]
                              -----
                              lockA            2            [<00000000603ef921>] thrd_work.cold+0x209/0x270
                                                                                 [deadlock_eg_AB_BA]

--
                       &mod->param_lock:           0              0           0.00            0.00           0.00
0.00               0              1             1.14           1.14           1.14            1.14
                              lockB:               0              0           0.00            0.00           0.00
0.00               4              6             8.06         213.96         410.10           68.35
                         cgroup_mutex:             0              0           0.00            0.00           0.00
0.00               2              3             4.44           4.82          14.04            4.68


con-bounces                                                 acq-bounces
number of lock contention that involved x-cpu data          number of lock acquisitions that involved x-cpu data

contentions                                                 acquisitions
number of lock acquisitions that had to wait                number of times we took the lock
```