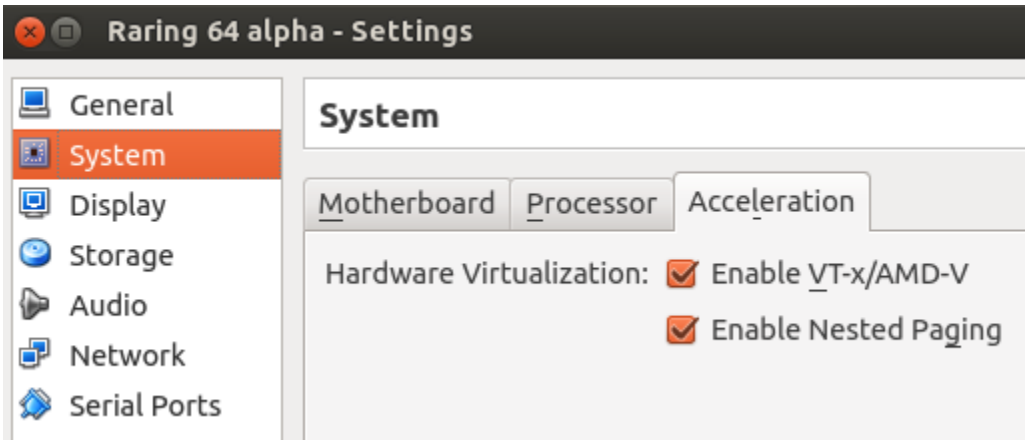
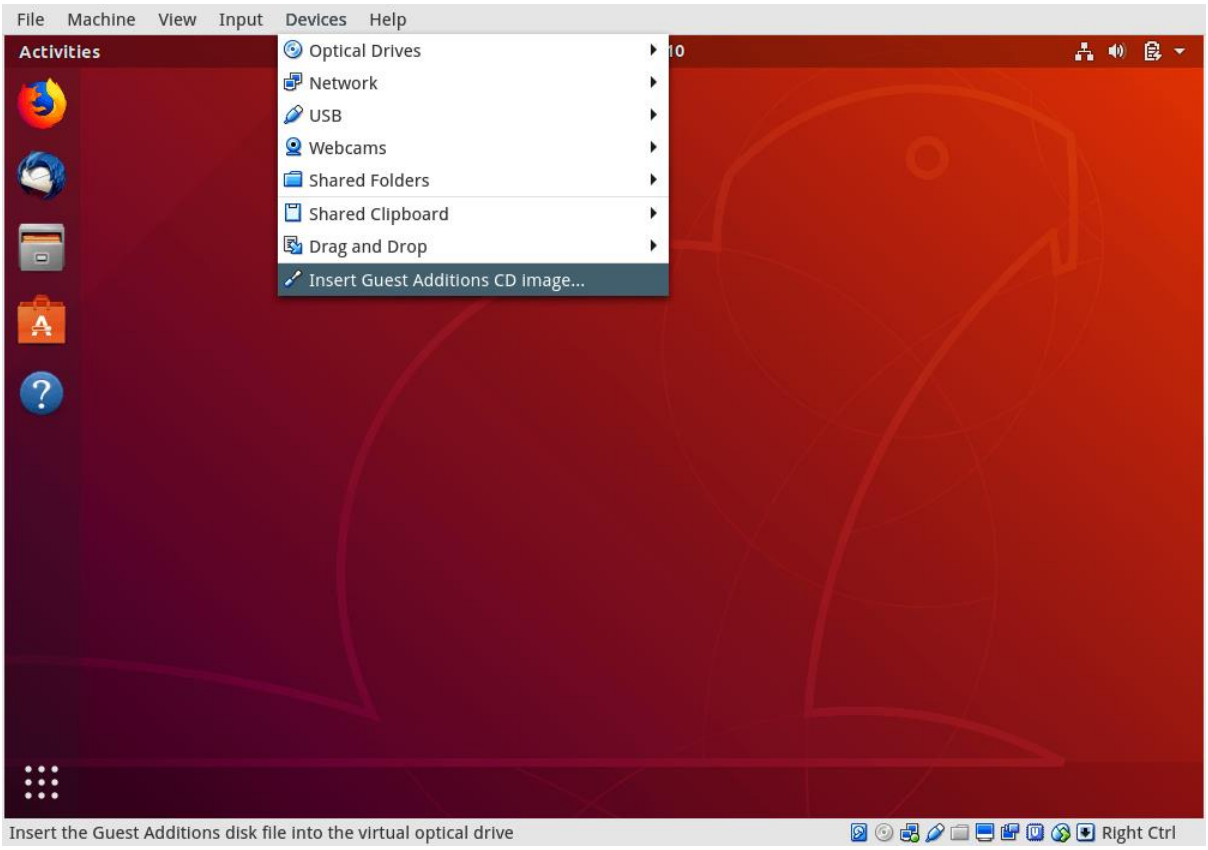
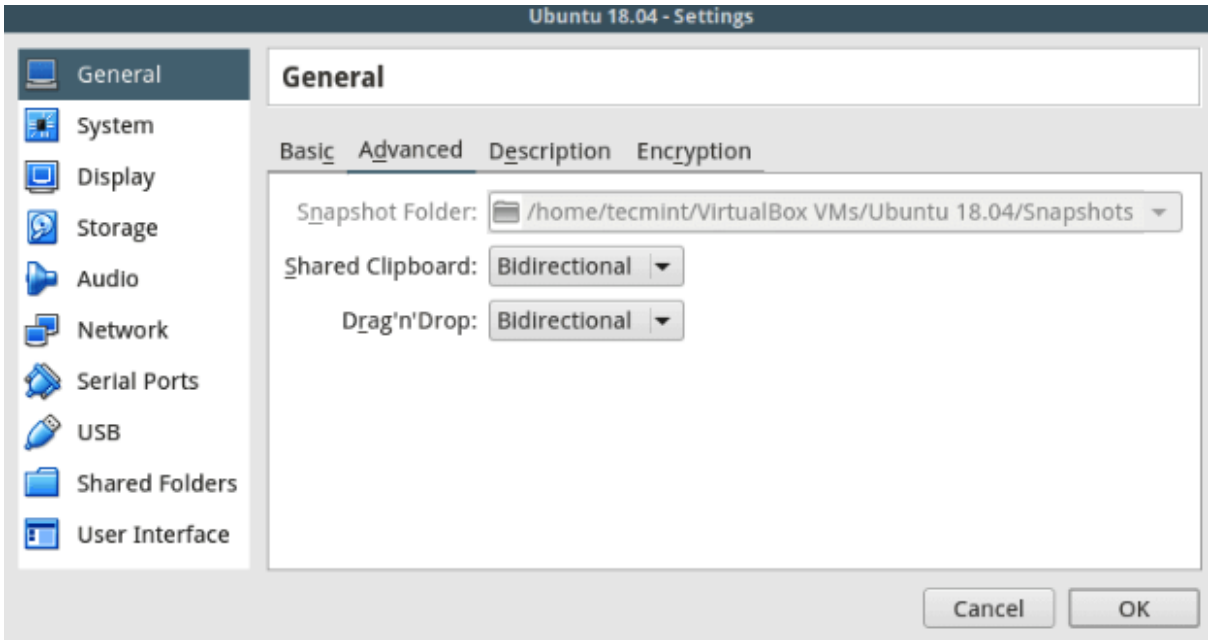
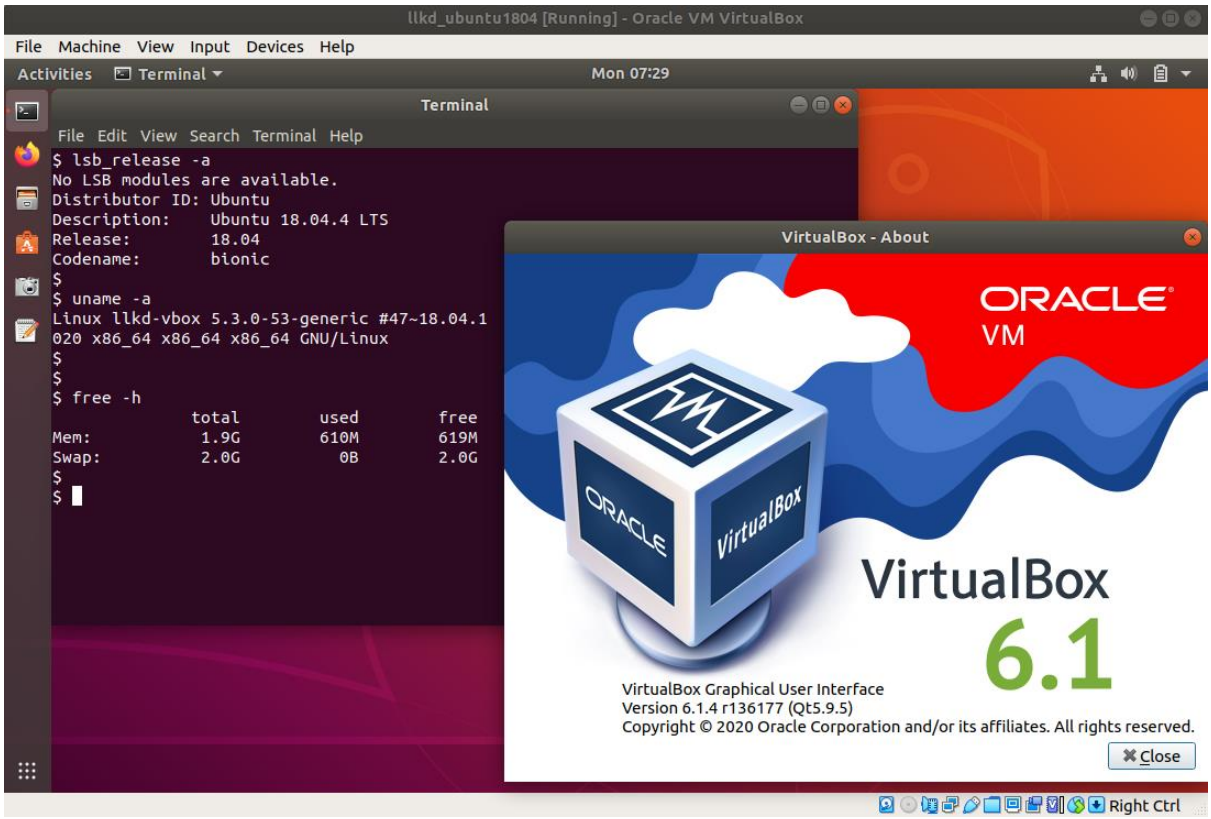
Chapter 1: Kernel Workspace Setup




File Machine View Input Devices Help
Activities
Optical Drives
Network
USB
Webcams
Shared Folders
Shared Clipboard
Drag and Drop
Insert Guest Additions CD image...
Insert the Guest Additions disk file into the virtual optical drive
Right Ctrl


Ubuntu 18.04 - Settings
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface
General
Basic
Advanced
Description
Encryption
Snapshot Folder:
/home/tecmint/VirtualBox VMs/Ubuntu 18.04/Snapshots
Shared Clipboard:
Bidirectional
Drag'n'Drop:
Bidirectional
Cancel
OK


llkd_ubuntu1804 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Mon 07:29
Terminal
File Edit View Search Terminal Help
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description: Ubuntu 18.04.4 LTS
Release: 18.04
Codename: bionic
$
$ uname -a
Linux llkd-vbox 5.3.0-53-generic #47~18.04.1
020 x86_64 x86_64 x86_64 GNU/Linux
$
$
$ free -h
total used free
Mem: 1.9G 610M 619M
Swap: 2.0G 0B 2.0G
$
$
VirtualBox - About
ORACLE
VM
ORACLE
VirtualBox
VirtualBox
6.1
VirtualBox Graphical User Interface
Version 6.1.4 r136177 (Qt5.9.5)
Copyright © 2020 Oracle Corporation and/or its affiliates. All rights reserved.
Close
Right Ctrl

```
~ $ tldr ps

  ps

  Information about running processes.

  - List all running processes:
    ps aux

  - List all running processes including the full command string:
    ps auxww

  - Search for a process that matches a string:
    ps aux | grep string

  - List all processes of the current user in extra full format:
    ps --user $(id -u) -F

  - List all processes of the current user as a tree:
    ps --user $(id -u) f

  - Get the parent pid of a process:
    ps -o ppid= -p pid


~ $
```

https://www.kernel.org/doc/html/latest/core-api/kernel-api.html#basic-c-library-functions

Kernel Maintainer Handbook
The Linux driver implementer's API guide
Core API Documentation
Core utilities
The Linux Kernel API
List Management Functions
Basic C Library Functions
Basic Kernel Library Functions
CRC and Math Functions in Linux
Kernel IPC facilities
FIFO Buffer
relay interface support
Module Support
Hardware Interfaces
Security Framework
Audit Interfaces
Accounting Framework
Block Devices
Char devices
Clock Framework
Synchronization Primitives

the name of the hlist_node within the struct.

Basic C Library Functions

When writing drivers, you cannot in general use routines which are from the C Library. Some of the functions have been found generally useful and they are listed below. The behaviour of these functions may vary slightly from those defined by ANSI, and these deviations are noted in the text.

String Conversions

unsigned long long simple_strtoull(const char * *cp*, char ** *endp*, unsigned int *base*)

convert a string to an unsigned long long

Parameters

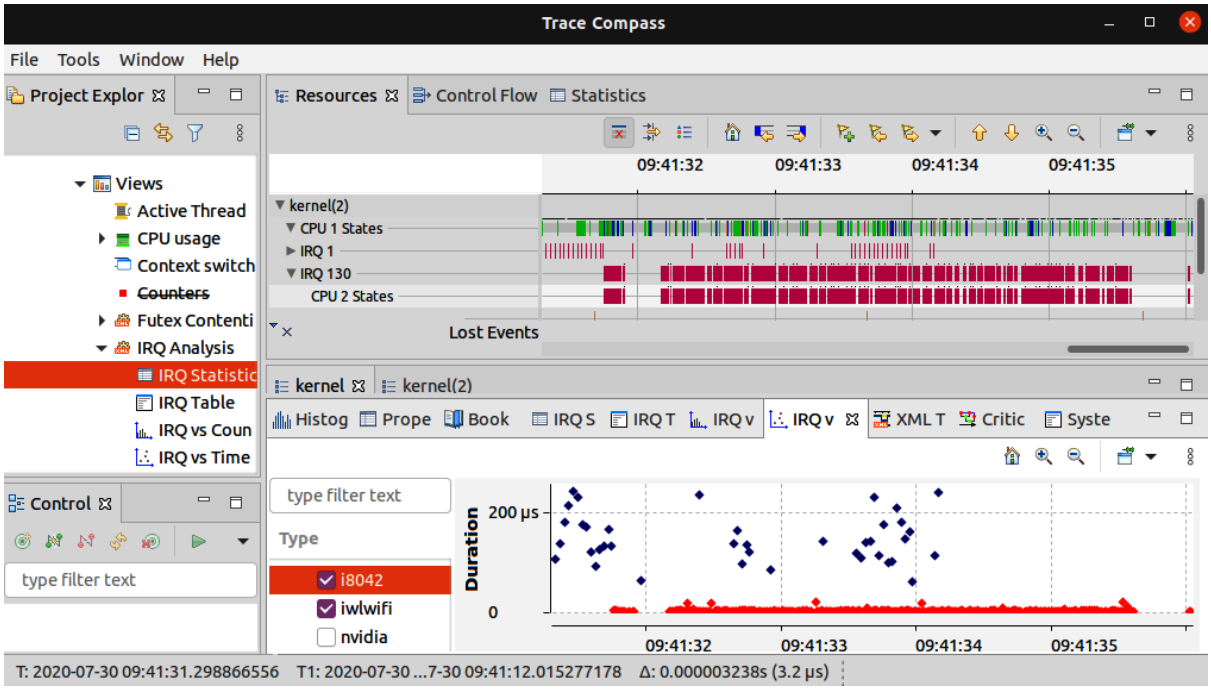
const char * cp
The start of the string

char ** endp
A pointer to the end of the parsed string will be placed here

unsigned int base
The number base to use

Description

This function is obsolete. Please use kstrtoull instead.



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

Sun Dec 27 09:47:44 IST 2020
[=====---  Start memory map for 1:systemd  ---======]
[Pathname: /usr/lib/systemd/systemd ]
VAS mappings:  name    [ size,perms,u:maptype,u:0xfile-offset]
+-----------------  K E R N E L   V A S    end kva  ------------------+ ffffffffffffffff
|<... K sparse region ...> [   8.00 MB,--- ]                          |
|                                                                     |
|                                                                     |
+---------------------------------------------------------------------+ ffffffffff7ff000
|       fixmap region [   2.52 MB,r-- ]                               |
|                                                                     |
|                                                                     |
+---------------------------------------------------------------------+ ffffffffff579000  <-- FIXADDR_START
|<... K sparse region ...> [   5.47 MB,--- ]                          |
|                                                                     |
|                                                                     |
+---------------------------------------------------------------------+ ffffffffff000000  <-- MODULES_END
|       module region [1008.00 MB,rwx ]                               |
|                                                                     |
|                                                                     |
|                                                                     |
|                                                                     |
|                                                                     |
|                                                                     |
+---------------------------------------------------------------------+ ffffffffc0000000  <-- MODULES_VADDR
|<... K sparse region ...> [  40.60 TB,--- ]                          |
|                                                                     |
|                                                                     |
|                                                                     |
|                                                                     |
|                                                                     |
|                                                                     |
|                                                                     |
|                                                                     |
~ .       .       .       .       .       .       .       .       .   ~
|                                                                     |
|                                                                     |
|                                                                     |
|                                                                     |
|                                                                     |
|                                                                     |
|                                                                     |
+---------------------------------------------------------------------+ ffffd764bfffffff  <-- VMALLOC_END
|      vmalloc region [  31.99 TB,rw- ]                               |
|                                                                     |
```

About Us
- About Center
- Our Team
- News
- Partners
- Contacts

Projects
- Linux Kernel Space Verification
- LSB Infrastructure
- Testing Technologies
- Tests and Frameworks
- Portability Tools

Results
- Contribution
- Publications
- Events

Online Linux Driver Verification Service (alpha)

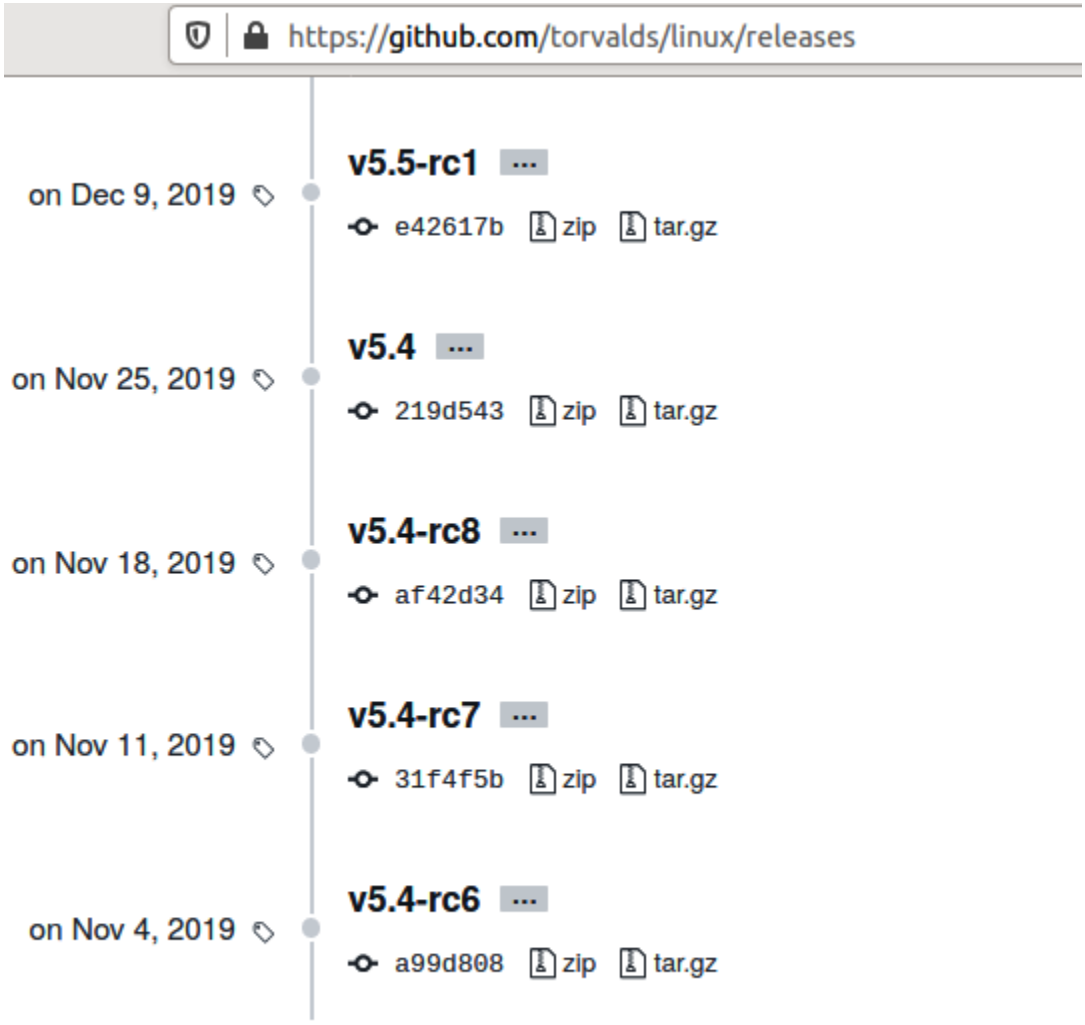
Start Verification Verification History Rules

Rules

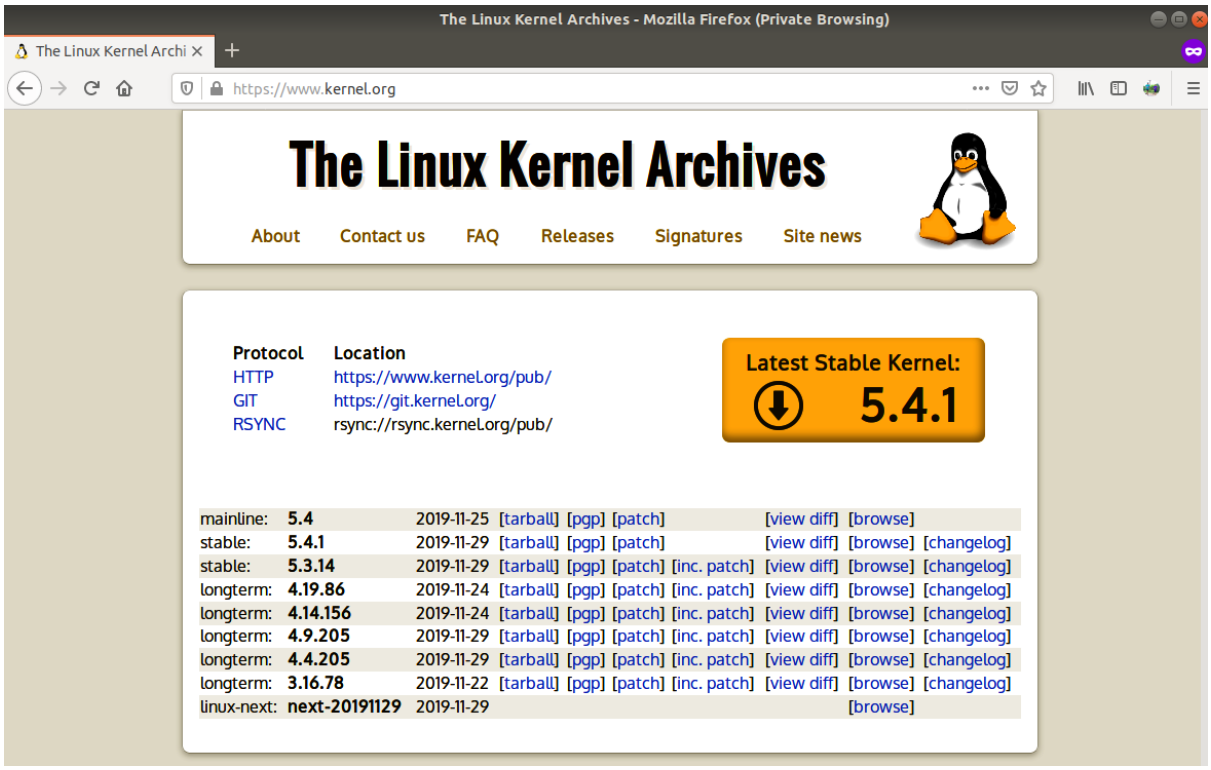
This page contains the list of verified rules. You can see more detailed information on them by clicking on the corresponding rule name.

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

Chapter 2: Building the 5.x Linux Kernel from Source - Part 1

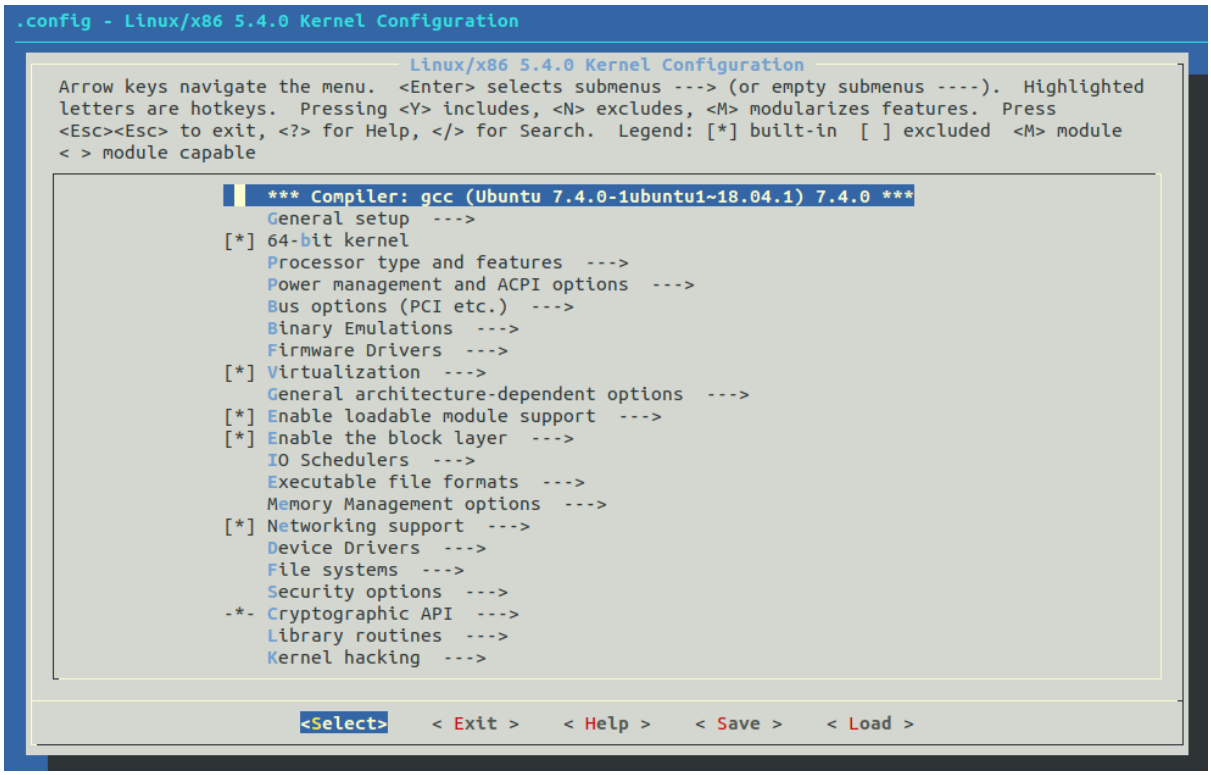


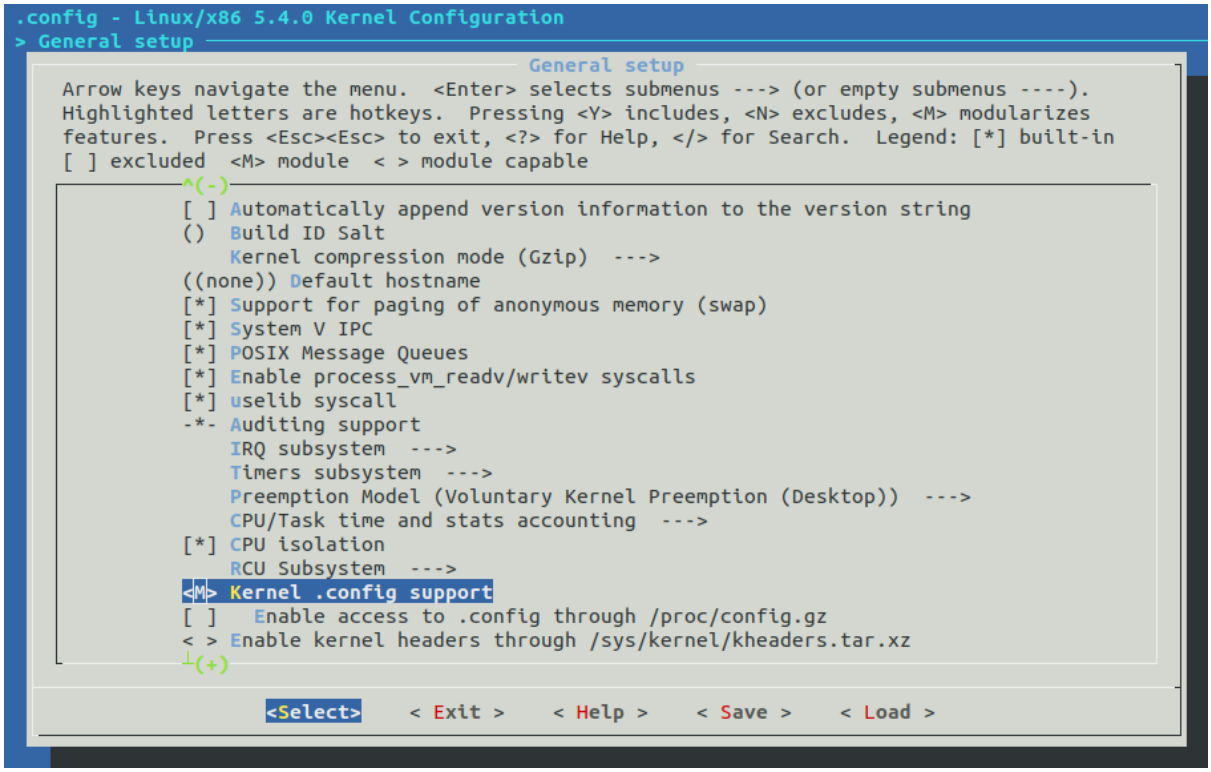
Previous Next

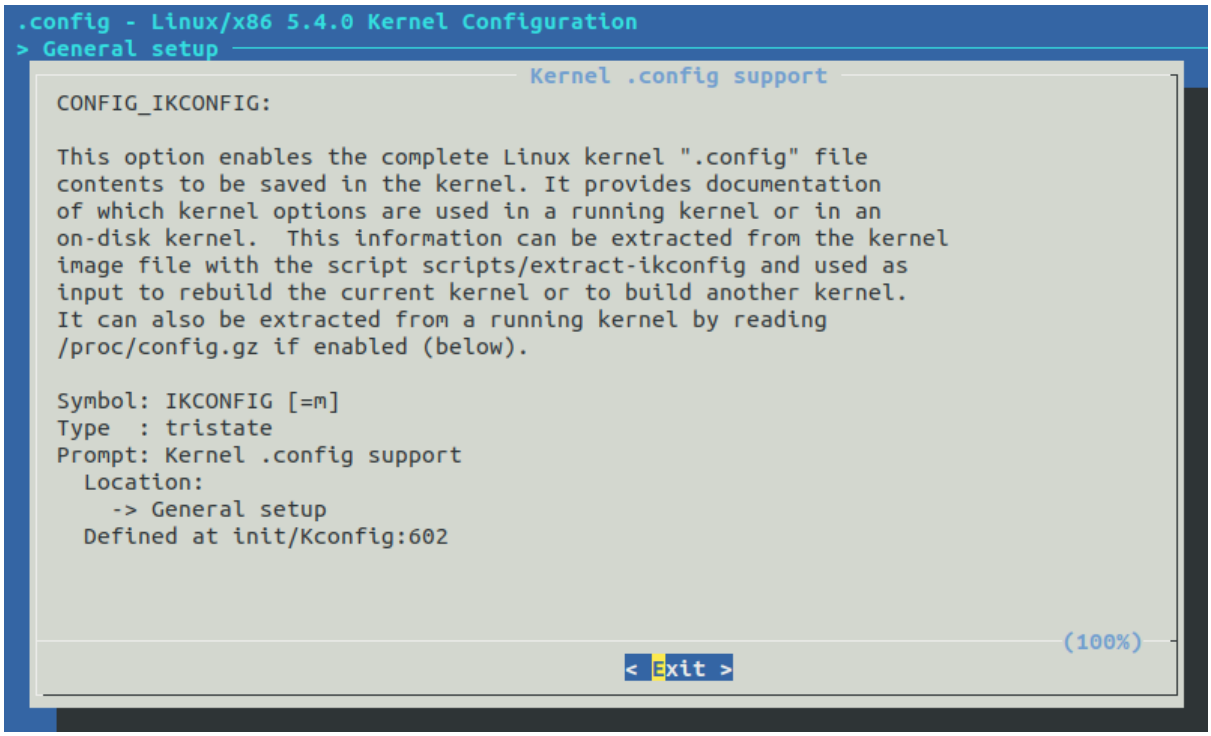



```
$ ls
arch/     crypto/          include/  kernel/      mm/        security/
block/    Documentation/   init/     lib/         net/       sound/
certs/    drivers/         ipc/      LICENSES/    README     tools/
COPYING   firmware/        Kbuild    MAINTAINERS  samples/   usr/
CREDITS   fs/              Kconfig   Makefile     scripts/   virt/
$
```

```
llkd linux-5.4 $ ls arch/arm/configs/
am200epdkit_defconfig     ezx_defconfig             multi_v5_defconfig    sama5_defconfig
aspeed_g4_defconfig       footbridge_defconfig      multi_v7_defconfig    shannon_defconfig
aspeed_g5_defconfig       gemini_defconfig          mv78xx0_defconfig     shmobile_defconfig
assabet_defconfig         h3600_defconfig           mvebu_v5_defconfig    simpad_defconfig
at91_dt_defconfig         h5000_defconfig           mvebu_v7_defconfig    socfpga_defconfig
axm55xx_defconfig         hackkit_defconfig         mxs_defconfig         spear13xx_defconfig
badge4_defconfig          hisi_defconfig            neponset_defconfig    spear3xx_defconfig
bcm2835_defconfig         imote2_defconfig          netwinder_defconfig   spear6xx_defconfig
cerfcube_defconfig        imx_v4_v5_defconfig       nhk8815_defconfig     spitz_defconfig
clps711x_defconfig        imx_v6_v7_defconfig       omap1_defconfig       stm32_defconfig
cm_x2xx_defconfig         integrator_defconfig      omap2plus_defconfig   sunxi_defconfig
cm_x300_defconfig         iop32x_defconfig          orion5x_defconfig     tango4_defconfig
cns3420vb_defconfig       ixp4xx_defconfig          oxnas_v6_defconfig    tct_hammer_defconfig
colibri_pxa270_defconfig  jornada720_defconfig      palmz72_defconfig     tegra_defconfig
colibri_pxa300_defconfig  keystone_defconfig        pcm027_defconfig      trizeps4_defconfig
collie_defconfig          lart_defconfig            pleb_defconfig        u300_defconfig
corgi_defconfig           lpc18xx_defconfig         prima2_defconfig      u8500_defconfig
davinci_all_defconfig     lpc32xx_defconfig         pxa168_defconfig      versatile_defconfig
dove_defconfig            lpd270_defconfig          pxa255-idp_defconfig  vexpress_defconfig
dram_0x00000000.config    lubbock_defconfig         pxa3xx_defconfig      vf610m4_defconfig
dram_0xc0000000.config    magician_defconfig        pxa910_defconfig      viper_defconfig
dram_0xd0000000.config    mainstone_defconfig       pxa_defconfig         vt8500_v6_v7_defconfig
ebsa110_defconfig         milbeaut_m10v_defconfig   qcom_defconfig        xcep_defconfig
efm32_defconfig           mini2440_defconfig        realview_defconfig    zeus_defconfig
em_x270_defconfig         mmp2_defconfig            rpc_defconfig         zx_defconfig
ep93xx_defconfig          moxart_defconfig          s3c2410_defconfig
eseries_pxa_defconfig     mps2_defconfig            s3c6400_defconfig
exynos_defconfig          multi_v4t_defconfig       s5pv210_defconfig
llkd linux-5.4 $
```




.config - Linux/x86 5.4.0 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
[ ] Automatically append version information to the version string
() Build ID Salt
Kernel compression mode (Gzip) --->
((none)) Default hostname
[*] Support for paging of anonymous memory (swap)
[*] System V IPC
[*] POSIX Message Queues
[*] Enable process_vm_readv/writev syscalls
[*] uselib syscall
-*- Auditing support
IRQ subsystem --->
Timers subsystem --->
Preemption Model (Voluntary Kernel Preemption (Desktop)) --->
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
<Select> < Exit > < Help > < Save > < Load >


.config - Linux/x86 5.4.0 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
Prompt: Kernel .config support
Location:
-> General setup
Defined at init/Kconfig:602
(100%)
< Exit >

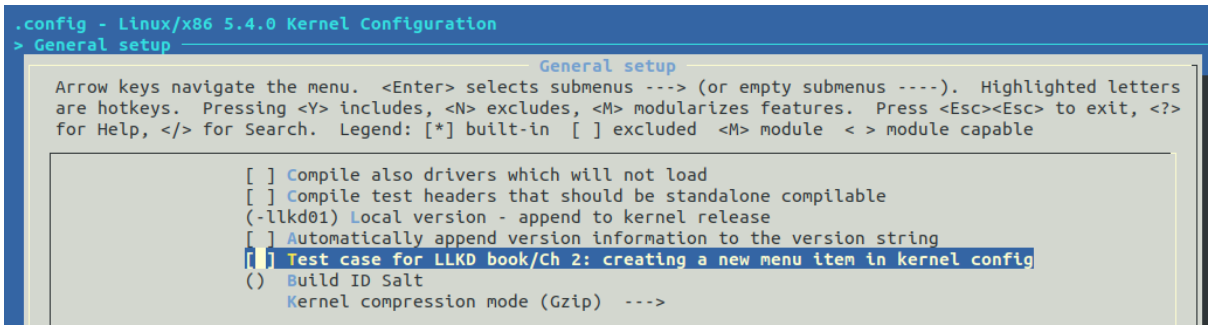
```
    RCU Subsystem  --->
<*> Kernel .config support
[*]   Enable access to .config through /proc/config.gz
(18) Kernel log buffer size (16 => 64KB, 17 => 128KB)
(12) CPU kernel log buffer size contribution (13 => 8 KB, 17
```

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

< Yes >      <  No  >
```

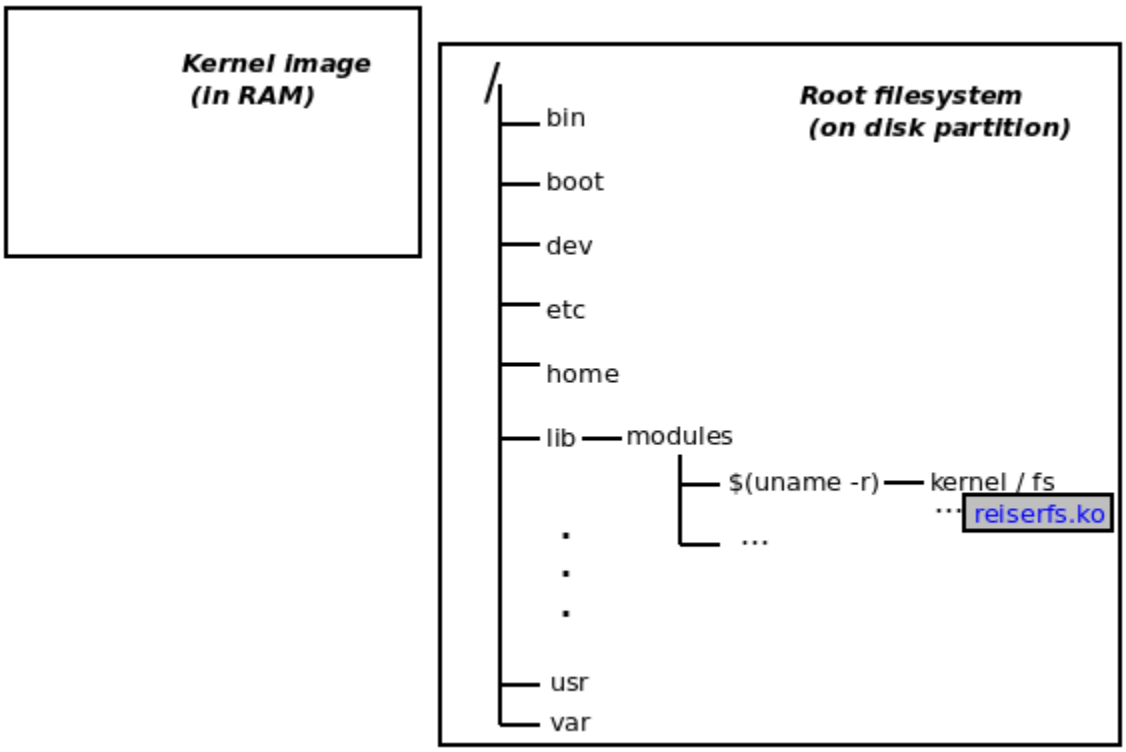
```
Search Configuration Parameter
Enter (sub)string or regexp to search for (with or without "CONFIG_")
vbox

< Ok >      < Help >
```

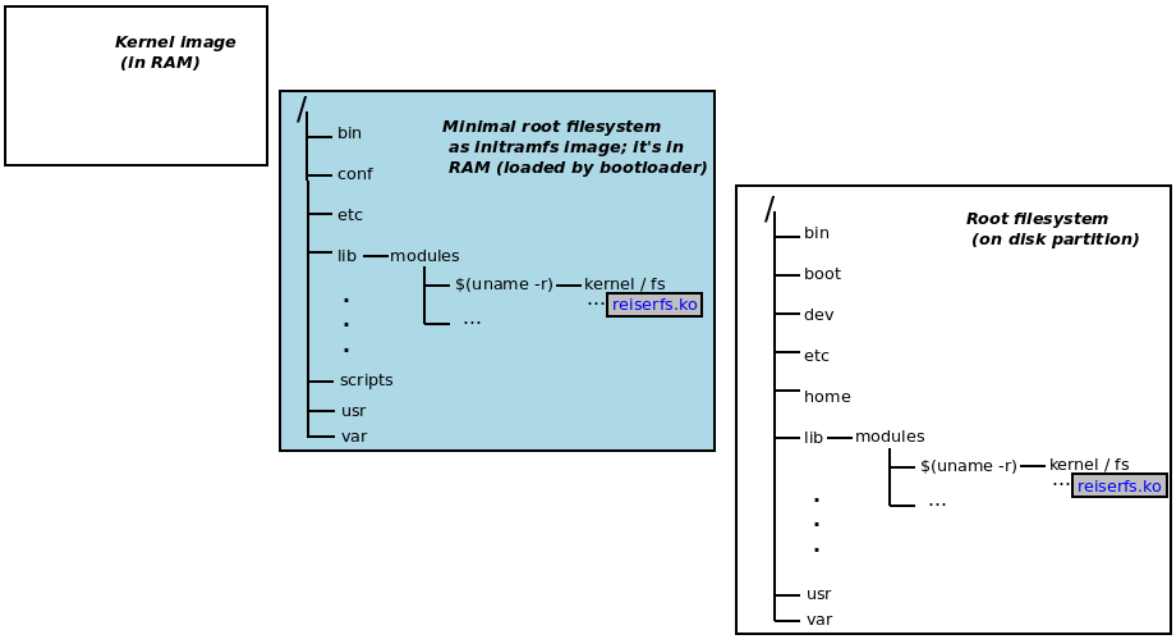
```
.config - Linux/x86 5.4.0 Kernel Configuration
> General setup > Search (vbox)
Search Results
Symbol: DRM_VBOXVIDEO [=m]
Type  : tristate
Prompt: Virtual Box Graphics Card
  Location:
    -> Device Drivers
(1)   -> Graphics support
  Defined at drivers/gpu/drm/vboxvideo/Kconfig:2
  Depends on: HAS_IOMEM [=y] && DRM [=m] && X86 [=y] && PCI [=y]
  Selects: DRM_KMS_HELPER [=m] && DRM_VRAM_HELPER [=m] && GENERIC_ALLOCATOR [=y]


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

(100%)
< Exit >
```

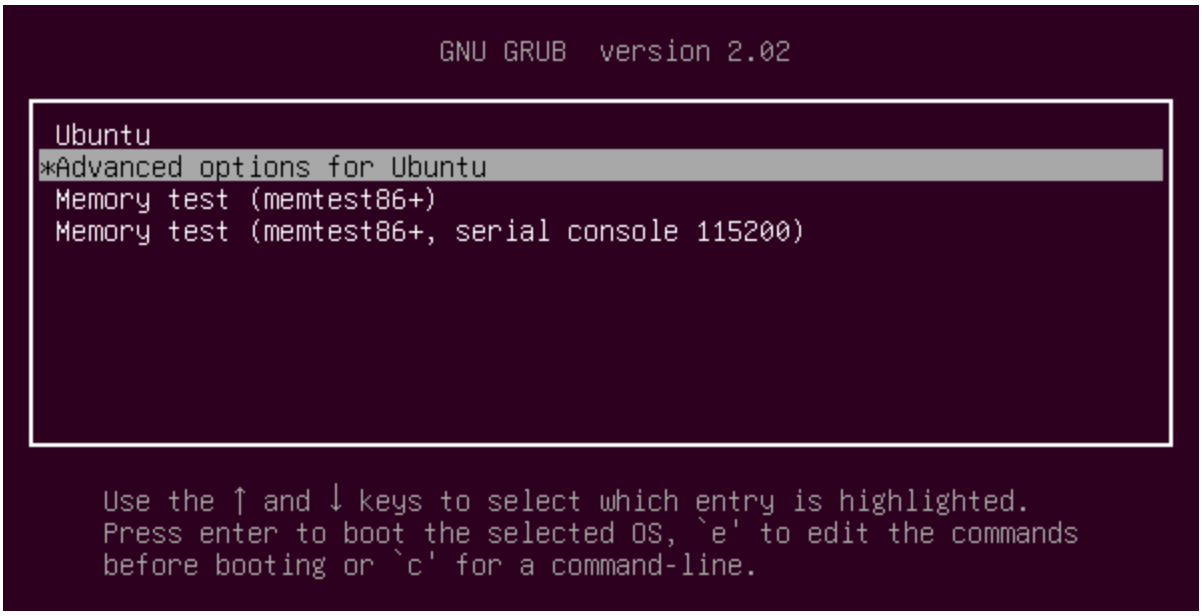
```
166       which is done within the script "scripts/setlocalversion".)
167
168 config LLKD_OPTION1
169        bool "Test case for LLKD book/Ch 2: creating a new menu item in kernel config"
170        default n
171        help
172           This option is merely a dummy 'test'; it's simply to have readers of our book
173           - 'Learn Linux Kernel Development', Kaiwan NB, Packt - try out the creation of
174           a few menu items within the kernel config.
175
176           Try setting this option to 'Y' (true), save and exit, and see the effect this
177           has by doing:
178            grep "CONFIG_LLKD_OPTION1" .config
179
180           If unsure, say N
181
182 config BUILD_SALT
183        string "Build ID Salt"
184        default ""
```

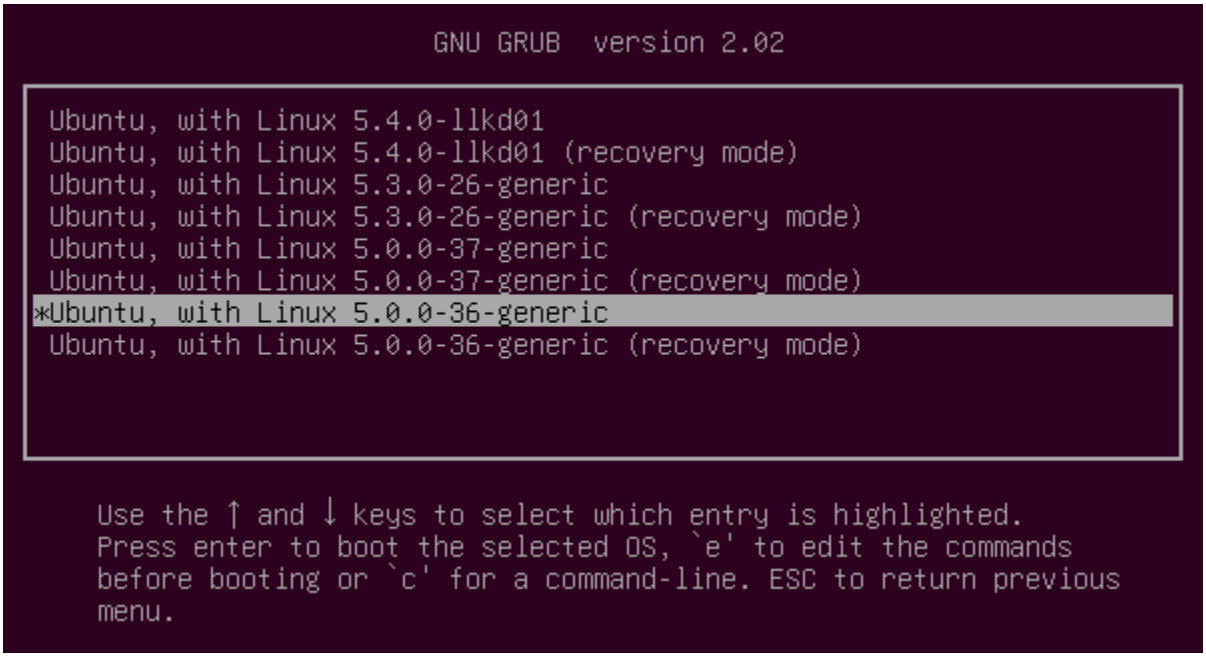


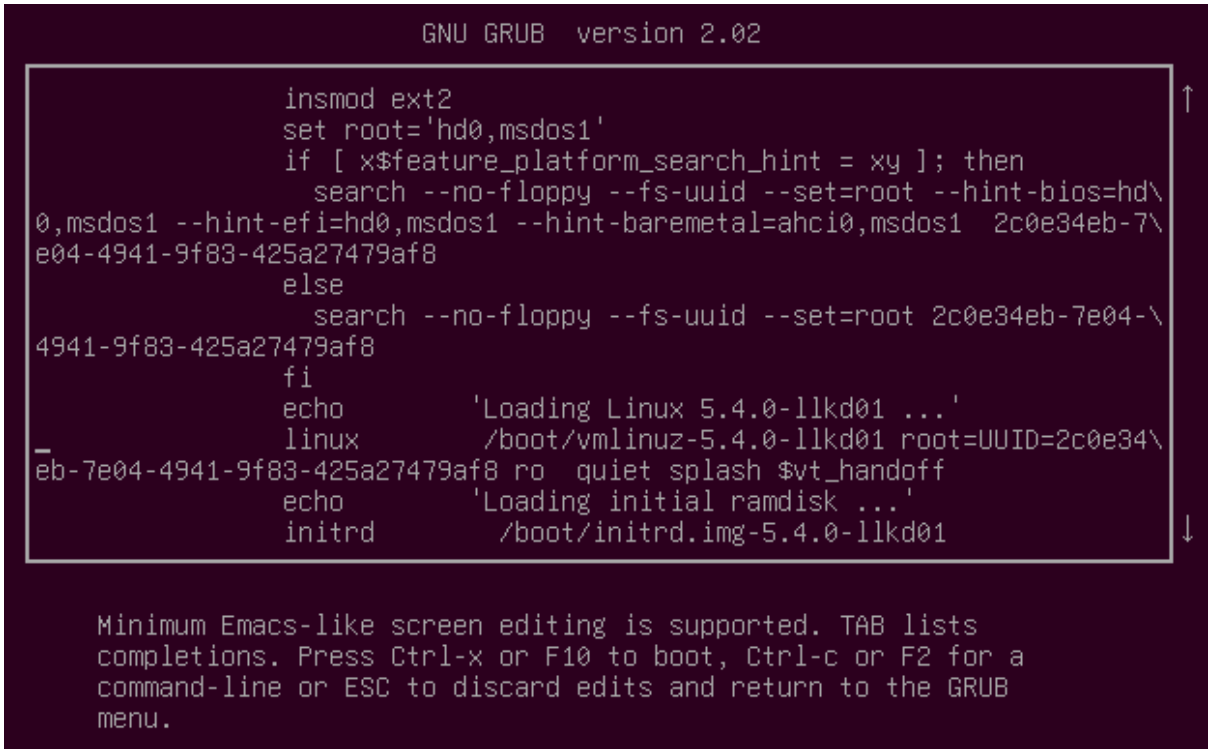
Chapter 3: Building the 5.x Linux Kernel from Source - Part 2

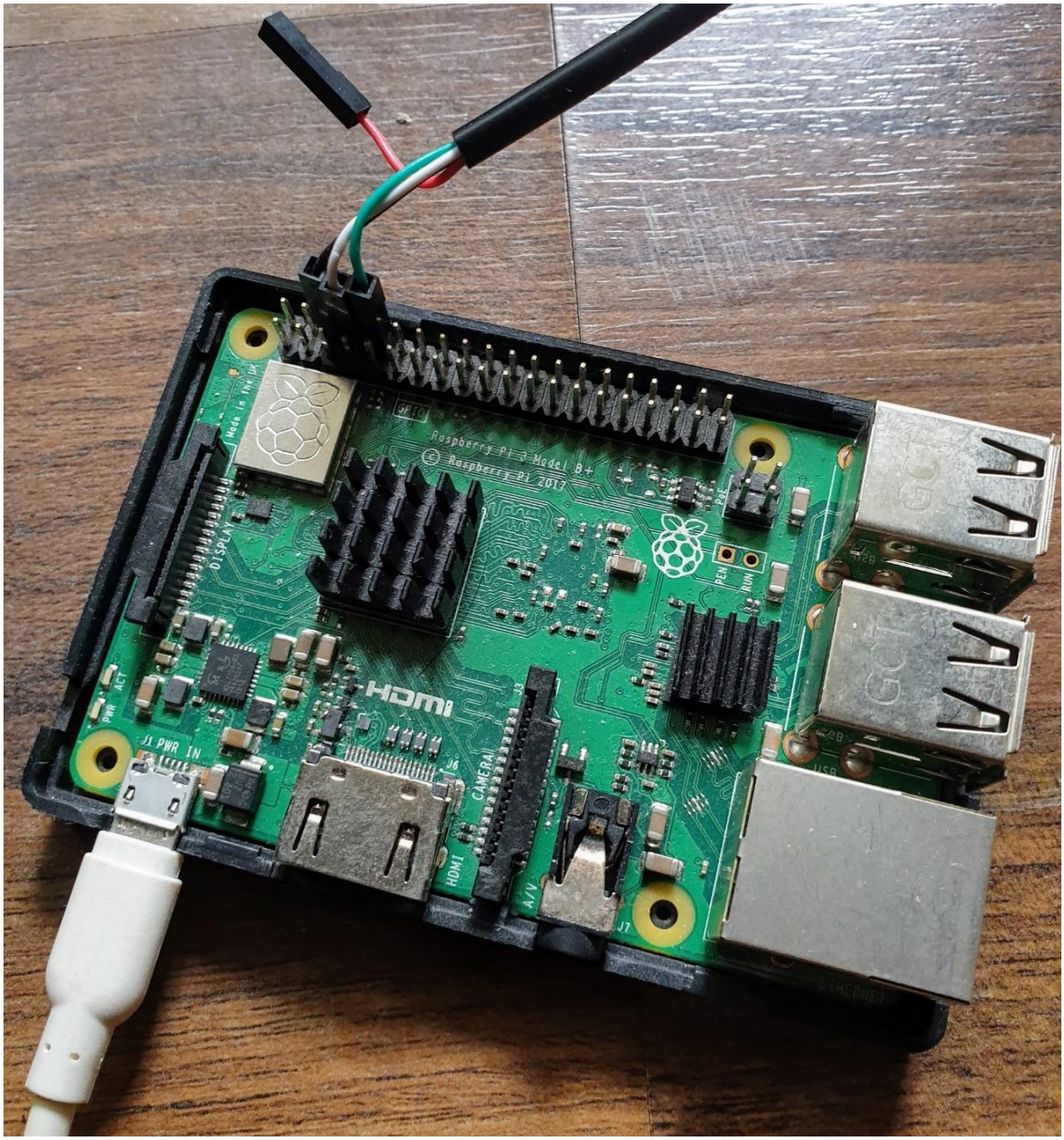




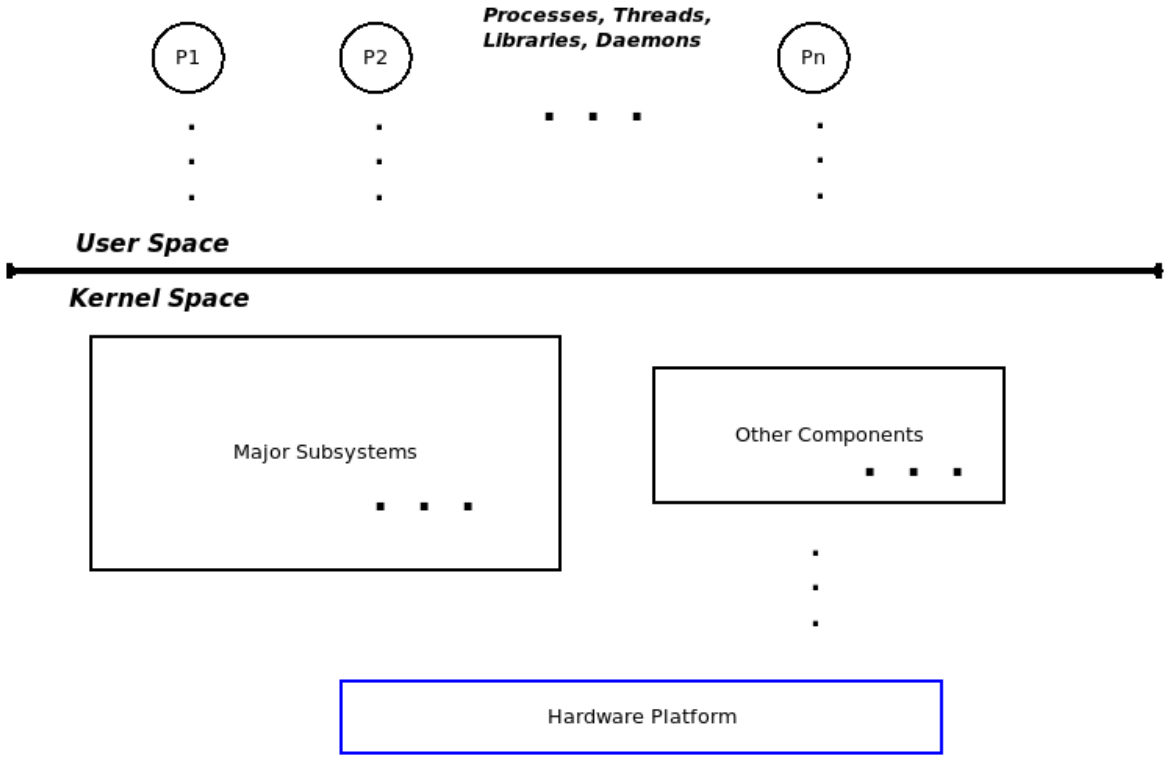

GNU GRUB version 2.02
Ubuntu
*Advanced options for Ubuntu
Memory test (memtest86+)
Memory test (memtest86+, serial console 115200)
Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, `e' to edit the commands
before booting or `c' for a command-line.

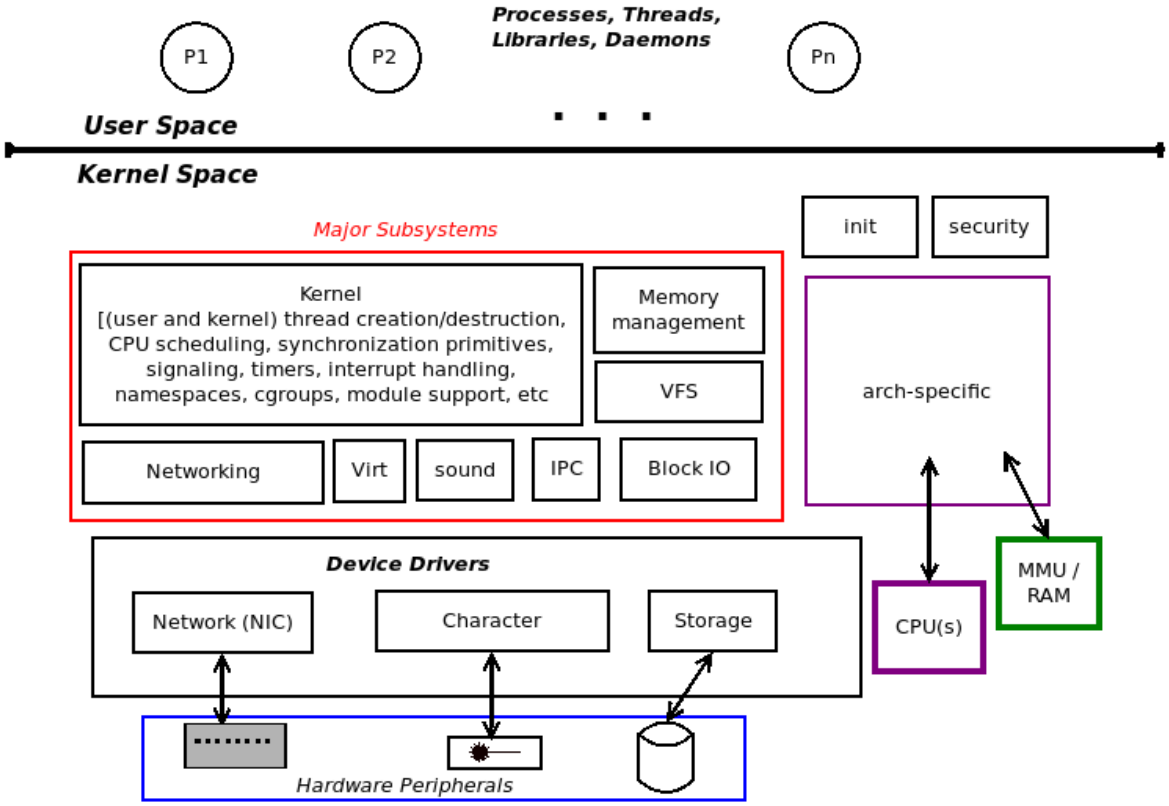

GNU GRUB version 2.02
Ubuntu, with Linux 5.4.0-llkd01
Ubuntu, with Linux 5.4.0-llkd01 (recovery mode)
Ubuntu, with Linux 5.3.0-26-generic
Ubuntu, with Linux 5.3.0-26-generic (recovery mode)
Ubuntu, with Linux 5.0.0-37-generic
Ubuntu, with Linux 5.0.0-37-generic (recovery mode)
*Ubuntu, with Linux 5.0.0-36-generic
Ubuntu, with Linux 5.0.0-36-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.

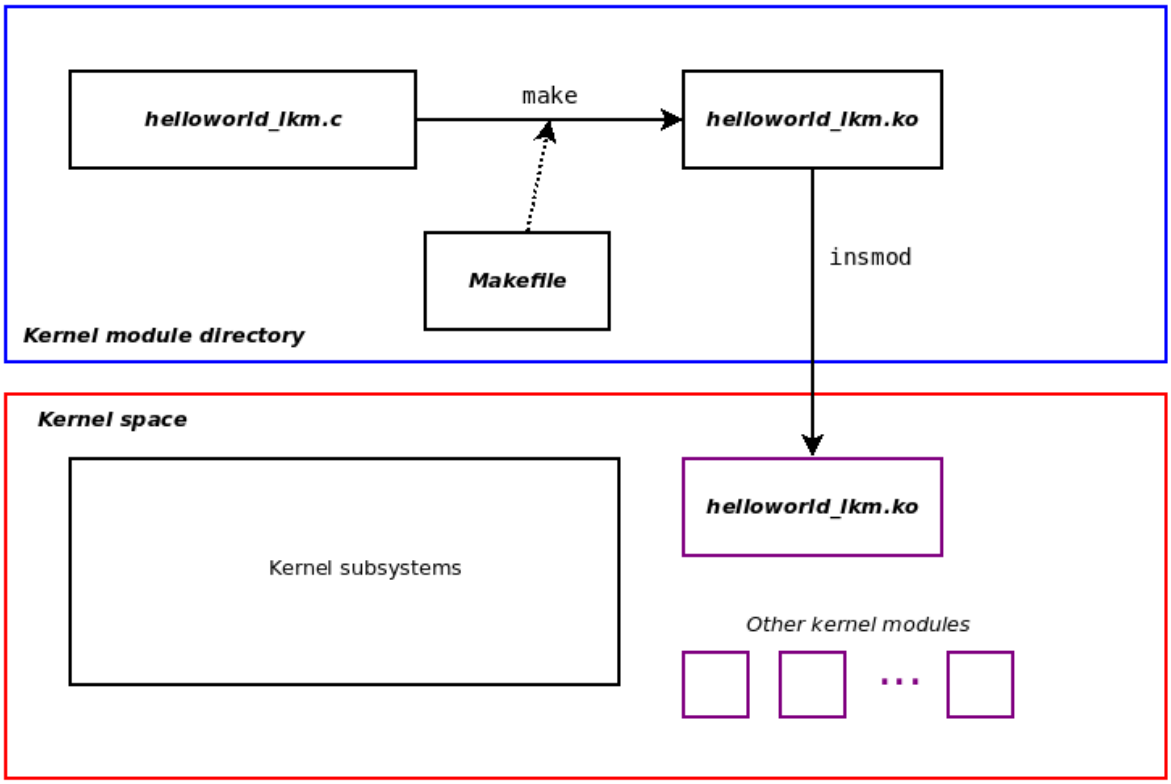

GNU GRUB version 2.02
insmod ext2
set root='hd0,msdos1'
if [x$feature_platform_search_hint = xy]; then
search --no-floppy --fs-uuid --set=root --hint-bios=hd\
0,msdos1 --hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 2c0e34eb-7\
e04-4941-9f83-425a27479af8
else
search --no-floppy --fs-uuid --set=root 2c0e34eb-7e04-\
4941-9f83-425a27479af8
fi
echo 'Loading Linux 5.4.0-llkd01 ...'
linux /boot/vmlinuz-5.4.0-llkd01 root=UUID=2c0e34\
eb-7e04-4941-9f83-425a27479af8 ro quiet splash $vt_handoff
echo 'Loading initial ramdisk ...'
initrd /boot/initrd.img-5.4.0-llkd01
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.


Raspberry Pi 3 Model B+
Raspberry Pi 2017
DISPLAY
HDMI
CAMERA
HDMI
A/V
J1 PWR IN
PWR
ACT
PoE
PEN
RUN
J7

Chapter 4: Writing Your First Kernel Module - LKMs Part 1




Processes, Threads,
Libraries, Daemons
P1
P2
Pn
User Space
Kernel Space
Major Subsystems
init
security
Kernel
[(user and kernel) thread creation/destruction,
CPU scheduling, synchronization primitives,
signaling, timers, interrupt handling,
namespaces, cgroups, module support, etc
Memory
management
VFS
arch-specific
Networking
Virt
sound
IPC
Block IO
Device Drivers
Network (NIC)
Character
Storage
CPU(s)
MMU /
RAM
Hardware Peripherals


helloworld_lkm.c
make
helloworld_lkm.ko
Makefile
insmod
Kernel module directory
Kernel space
Kernel subsystems
helloworld_lkm.ko
Other kernel modules

```
llkd ~ $ ls /lib/modules/5.0.0-36-generic/kernel/drivers/net/ethernet/
3com/        amd/       chelsio/   ethoc.ko   mellanox/   ni/            samsung/   tehuti/
8390/        aquantia/  cirrus/    fealnx.ko  micrel/     nvidia/        sfc/       ti/
adaptec/     atheros/   cisco/     fujitsu/   microchip/  packetengines/ silan/     via/
agere/       aurora/    dec/       hp/        mscc/       qlogic/        sis/       wiznet/
alacritech/  broadcom/  dlink/     huawei/    myricom/    qualcomm/      smsc/      xircom/
alteon/      brocade/   dnet.ko    intel/     natsemi/    rdc/           stmicro/
altera/      cadence/   ec_bhf.ko  jme.ko     neterion/   realtek/       sun/
amazon/      cavium/    emulex/    marvell/   netronome/  rocker/        synopsys/
llkd ~ $
```

```
$ ls -l
total 8
-rw-rw-r-- 1 llkd llkd 1211 Jan 24 13:05 helloworld_lkm.c
-rw-rw-r-- 1 llkd llkd  333 Jan 24 13:05 Makefile
$ make
make -C /lib/modules/5.4.0-llkd01/build/ M=/home/llkd/llkd_book/Learn-Linux-Kernel-
Development/ch4/helloworld_lkm modules
make[1]: Entering directory '/home/llkd/kernels/linux-5.4'
  CC [M]  /home/llkd/llkd_book/Learn-Linux-Kernel-Development/ch4/helloworld_lkm/he
lloworld_lkm.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/llkd/llkd_book/Learn-Linux-Kernel-Development/ch4/helloworld_lkm/he
lloworld_lkm.mod.o
  LD [M]  /home/llkd/llkd_book/Learn-Linux-Kernel-Development/ch4/helloworld_lkm/he
lloworld_lkm.ko
make[1]: Leaving directory '/home/llkd/kernels/linux-5.4'
$ ls -l helloworld_lkm.ko
-rw-rw-r-- 1 llkd llkd 217224 Mar 17 17:29 helloworld_lkm.ko
$
```

```
$ lsb_release -a|grep Description
Description:    CentOS Linux release 8.0.1905 (Core)
$ uname -r
5.4.0-llkd01
$ ls
helloworld_lkm.c  Makefile
$ make
make -C /lib/modules/5.4.0-llkd01/build/ M=/home/llkd/bookwork/Learn-Linux-Kernel-Development/ch4/helloworld_lkm
 modules
make[1]: Entering directory '/home/llkd/bookwork/linux-5.4'
  CC [M]  /home/llkd/bookwork/Learn-Linux-Kernel-Development/ch4/helloworld_lkm/helloworld_lkm.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/llkd/bookwork/Learn-Linux-Kernel-Development/ch4/helloworld_lkm/helloworld_lkm.mod.o
  LD [M]  /home/llkd/bookwork/Learn-Linux-Kernel-Development/ch4/helloworld_lkm/helloworld_lkm.ko
make[1]: Leaving directory '/home/llkd/bookwork/linux-5.4'
$ ls -l ./helloworld_lkm.ko
-rw-rw-r-- 1 llkd llkd 202592 Nov 27 18:24 ./helloworld_lkm.ko
$ sudo insmod ./helloworld_lkm.ko
$ dmesg |tail -n1
[ 4731.967653] Hello, world
$ lsmod |grep helloworld_lkm
helloworld_lkm         16384  0
$ sudo rmmod helloworld_lkm
$ dmesg |tail -n2
[ 4731.967653] Hello, world
[ 4767.651584] Goodbye, world
$
```

```
rpi #
rpi # cat /proc/sys/kernel/printk
3       4       1       3
rpi #
rpi # insmod ./printk_loglvl.ko
[  257.712077] Hello, world @ log-level KERN_EMERG   [0]
[  257.719735] Hello, world @ log-level KERN_ALERT   [1]
[  257.727371] Hello, world @ log-level KERN_CRIT    [2]
rpi #
Message from syslogd@raspberrypi at Dec 17 05:36:01 ...
 kernel:[  257.712077] Hello, world @ log-level KERN_EMERG   [0]
```

```
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Online 3:5 | ttyUSB0
```

```
rpi #
rpi # cat /proc/sys/kernel/printk
3       4       1       3
rpi # echo "8 4 1 3" > /proc/sys/kernel/printk
rpi # cat /proc/sys/kernel/printk
8       4       1       3
rpi # rmmod printk_loglvl
[  481.197569] Goodbye, world @ log-level KERN_INFO    [6]
rpi # insmod ./printk_loglvl.ko
[  488.427733] Hello, world @ log-level KERN_EMERG   [0]
[  488.435585] Hello, world @ log-level KERN_ALERT   [1]
[  488.443264] Hello, world @ log-level KERN_CRIT    [2]
[  488.450865] Hello, world @ log-level KERN_ERR     [3]
rpi # [  488.450868] Hello, world @ log-level KERN_WARNING [4]
[  488.450870] Hello, world @ log-level KERN_NOTICE  [5]
[  488.450873] Hello, world @ log-level KERN_INFO    [6]

Message from syslogd@raspberrypi at Dec 17 05:39:52 ...
 kernel:[  488.427733] Hello, world @ log-level KERN_EMERG   [0]
```

```
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Online 3:5 | ttyUSB0
```

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

Message from syslogd@raspberrypi at Mar 18 11:37:15 ...
 kernel:[  975.271766] Hello, world @ log-level KERN_EMERG   [0]
printk_loglvl          16384  0
------------------------------
dmesg
------------------------------
[  975.271766] Hello, world @ log-level KERN_EMERG   [0]
[  975.277729] Hello, world @ log-level KERN_ALERT   [1]
[  975.283662] Hello, world @ log-level KERN_CRIT    [2]
[  975.289561] Hello, world @ log-level KERN_ERR     [3]
[  975.295394] Hello, world @ log-level KERN_WARNING [4]
[  975.301176] Hello, world @ log-level KERN_NOTICE  [5]
[  975.306907] Hello, world @ log-level KERN_INFO    [6]
[  975.312625] Hello, world @ log-level KERN_DEBUG   [7]
[  975.312628] Hello, world via the pr_devel() macro (eff @KERN_DEBUG) [7]
```

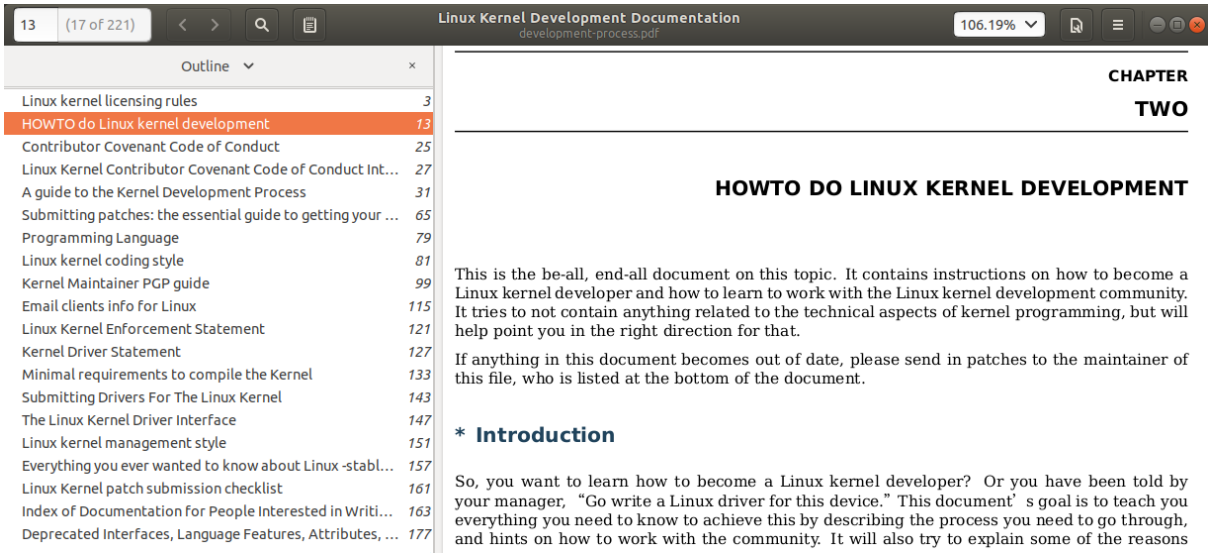
Chapter 5: Writing Your First Kernel Module - LKMs Part 2

```
lkm_template $ make
all            clean          help           install        sa_cppcheck    sa_gcc         tarxz-pkg
checkpatch     code-style     indent         sa             sa_flawfinder  sa_sparse
lkm_template $ make help
=== Makefile Help : additional targets available ===

TIP: type make <tab><tab> to show all valid targets

--- 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/5.4.0-58-generic/)
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 (requires spatch): https://www.kernel.org/doc/html/v4.15/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; requires conf
iguring 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 locking is
sues 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 test 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 the modul
e on another system
 Tip: when extracting, to extract into a dir of the same name as the tar file,
  do: tar -xvf lkm_template.tar.xz --one-top-level
help       : this help target
lkm_template $
```

```
rpi $ cat /proc/version
Linux version 5.4.51-v7+ (kaiwan@kaiwan-T460) (gcc version 4.8.3 20140303 (prerelease) (crosstool-NG
linaro-1.13.1+bzr2650 - Linaro GCC 2014.03)) #1 SMP Thu Jul 23 12:36:25 IST 2020
rpi $
rpi $ modinfo ./helloworld_lkm.ko
filename:       /home/pi/booksrc/ch5/cross/./helloworld_lkm.ko
version:        0.1
license:        Dual MIT/GPL
description:    LLKD book:ch5/cross: hello, world, our first Raspberry Pi LKM
author:         Kaiwan N Billimoria
srcversion:     7DDCE78A55CF6EDEEE783FF
depends:
name:           helloworld_lkm
vermagic:       5.4.51-v7+ SMP mod_unload modversions ARMv7 p2v8
rpi $ sudo dmesg -C
rpi $ sudo rmmod helloworld_lkm 2>/dev/null
rpi $ sudo insmod ./helloworld_lkm.ko
rpi $ dmesg
[ 3302.140940] Hello, Raspberry Pi world
rpi $ lsmod |grep helloworld_lkm
helloworld_lkm         16384  0
rpi $ sudo rmmod helloworld_lkm 2>/dev/null
rpi $ dmesg
[ 3302.140940] Hello, Raspberry Pi world
[ 3312.406669] Goodbye, Raspberry Pi world
rpi $
```

```
lttng_probe_irq        16384  0
lttng_probe_gpio       16384  0
lttng_probe_compaction    16384  0
lttng_probe_block      36864  0
lttng_probe_asoc       24576  0
lttng_ring_buffer_metadata_mmap_client    16384  0
lttng_ring_buffer_client_mmap_overwrite    20480  0
lttng_ring_buffer_client_mmap_discard    20480  0
lttng_ring_buffer_metadata_client    16384  0
lttng_ring_buffer_client_overwrite    20480  0
lttng_ring_buffer_client_discard    20480  0
lttng_tracer         1523712  35 lttng_probe_udp,lttng_probe_scsi,lttng_probe_sched,lttng_probe_compaction,lt
tng_probe_net,lttng_probe_vmscan,lttng_probe_writeback,lttng_probe_power,lttng_probe_rcu,lttng_probe_module,l
ttng_ring_buffer_client_mmap_overwrite,lttng_probe_statedump,lttng_ring_buffer_client_discard,lttng_probe_pri
ntk,lttng_probe_sock,lttng_probe_asoc,lttng_probe_irq,lttng_ring_buffer_client_mmap_discard,lttng_probe_kvm,l
ttng_probe_random,lttng_probe_timer,lttng_probe_workqueue,lttng_probe_jbd2,lttng_probe_v4l2,lttng_probe_signa
l,lttng_probe_skb,lttng_probe_block,lttng_probe_napi,lttng_ring_buffer_metadata_client,lttng_probe_kmem,lttng
_ring_buffer_metadata_mmap_client,lttng_probe_gpio,lttng_ring_buffer_client_overwrite,lttng_probe_regulator,l
ttng_probe_sunrpc
lttng_statedump       737280  1 lttng_tracer
lttng_kprobes          16384  1 lttng_tracer
lttng_clock            16384  5 lttng_ring_buffer_client_mmap_overwrite,lttng_ring_buffer_client_discard,lttn
g_tracer,lttng_ring_buffer_client_mmap_discard,lttng_ring_buffer_client_overwrite
lttng_lib_ring_buffer    57344  23 lttng_probe_scsi,lttng_probe_sched,lttng_probe_net,lttng_probe_power,lttng
_probe_rcu,lttng_probe_module,lttng_ring_buffer_client_mmap_overwrite,lttng_probe_statedump,lttng_ring_buffer
_client_discard,lttng_probe_printk,lttng_probe_sock,lttng_tracer,lttng_probe_asoc,lttng_probe_irq,lttng_ring_
buffer_client_mmap_discard,lttng_probe_kvm,lttng_probe_random,lttng_probe_napi,lttng_ring_buffer_metadata_cli
ent,lttng_ring_buffer_metadata_mmap_client,lttng_ring_buffer_client_overwrite,lttng_probe_regulator,lttng_pro
be_sunrpc
lttng_kretprobes       16384  1 lttng_tracer
~ $
```

```
------------------------------
dmesg
------------------------------
[ 5732.199769] fp_in_lkm: inserted
[ 5732.200659] fp_in_lkm: PI =
[ 5732.200666] ------------[ cut here ]------------
[ 5732.200667] Please remove unsupported %f in format string
[ 5732.200667] WARNING: CPU: 1 PID: 3524 at lib/vsprintf.c:2366 format_decode+0x3f4/0x400
[ 5732.200668] Modules linked in: fp_in_lkm(OE+) vboxsf(OE) vboxvideo(OE) vmwgfx drm_kms_helper syscopyarea snd_intel8x0 sysfillrect snd_ac97_
codec crct10dif_pclmul sysimgblt crc32_pclmul fb_sys_fops ac97_bus ghash_clmulni_intel ttm snd_pcm aesni_intel glue_helper drm snd_seq crypto_
simd joydev cryptd snd_timer snd_seq_device input_leds intel_rapl_perf snd serio_raw soundcore vboxguest(OE) video mac_hid sch_fq_codel min_sy
sinfo(O) parport_pc ppdev lp parport ip_tables x_tables autofs4 hid_generic usbhid hid ahci e1000 psmouse libahci i2c_piix4 pata_acpi
[ 5732.200683] CPU: 1 PID: 3524 Comm: insmod Tainted: G           OE     5.4.0-llkd01 #1
[ 5732.200683] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[ 5732.200683] RIP: 0010:format_decode+0x3f4/0x400
[ 5732.200685] Code: ff ff 48 8d 42 02 b9 4c 00 00 00 48 89 45 e8 e9 cd fc ff ff 0f be f2 48 c7 c7 68 e3 3b a0 c6 05 5e 4b c1 00 01 e8 2c 95 6
d ff <0f> 0b 48 8b 45 e8 e9 ff fe ff ff 90 89 f0 c1 e0 08 c1 f8 08 89 c2
[ 5732.200685] RSP: 0018:ffffb57fc1ae3a70 EFLAGS: 00010086
[ 5732.200686] RAX: 0000000000000000 RBX: ffffb57fc1ae3ab0 RCX: 0000000000000000
[ 5732.200687] RDX: 0000000000000003 RSI: ffffffffa03be398 RDI: ffff987dfdb19488
[ 5732.200687] RBP: ffffb57fc1ae3a88 R08: 0000000000000000 R09: ffffffffa0b30600
[ 5732.200688] R10: 00000000a0b30221 R11: 00000000ffffffff R12: ffffffffc0600047
[ 5732.200689] R13: 00000000000003e0 R14: ffffffffc0600047 R15: ffffffffc0600047
[ 5732.200689] FS:  00007fa67a2ce540(0000) GS:ffff987dfdb00000(0000) knlGS:0000000000000000
[ 5732.200690] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[ 5732.200690] CR2: 0000560751f8f788 CR3: 000000005f560005 CR4: 0000000000606e0
[ 5732.200690] Call Trace:
[ 5732.200691]  vsnprintf+0x66/0x510
[ 5732.200691]  vscnprintf+0xd/0x30
[ 5732.200692]  vprintk_store+0x3e/0x220
[ 5732.200692]  ? vprintk_func+0x47/0xc0
[ 5732.200692]  vprintk_emit+0xa9/0x2d0
[ 5732.200692]  ? 0xffffffffc0605000
[ 5732.200693]  vprintk_default+0x29/0x50
[ 5732.200694]  vprintk_func+0x47/0xc0
[ 5732.200694]  printk+0x52/0x6e
[ 5732.200694]  fp_in_lkm_init+0x5e/0x1000 [fp_in_lkm]
[ 5732.200695]  do_one_initcall+0x4a/0x1fa
```



CHAPTER

TWO

HOWTO DO LINUX KERNEL DEVELOPMENT

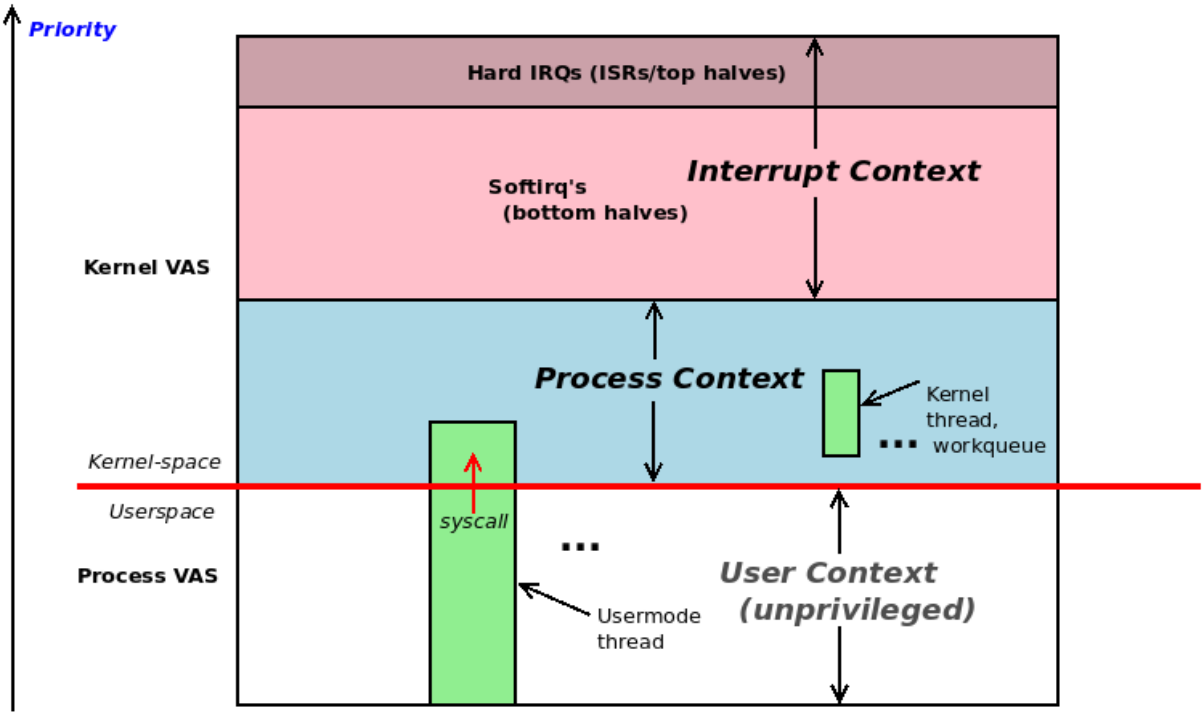
This is the be-all, end-all document on this topic. It contains instructions on how to become a Linux kernel developer and how to learn to work with the Linux kernel development community. It tries to not contain anything related to the technical aspects of kernel programming, but will help point you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

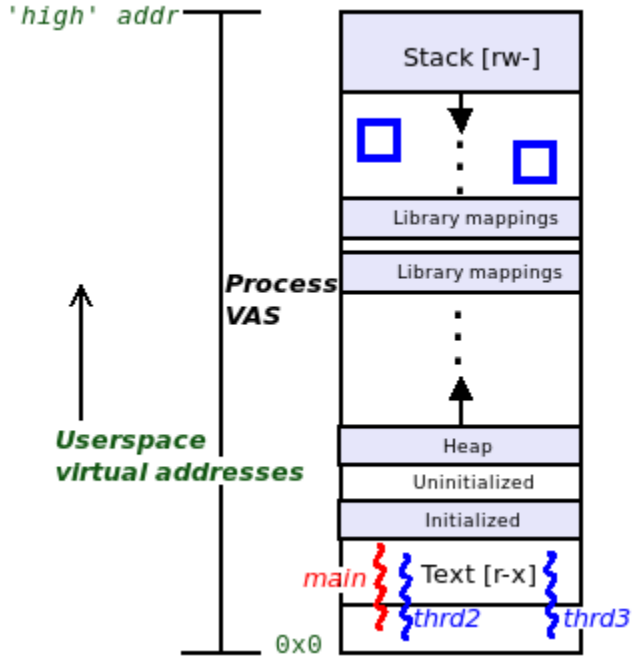
If anything in this document becomes out of date, please send in patches to the maintainer of this file, who is listed at the bottom of the document.

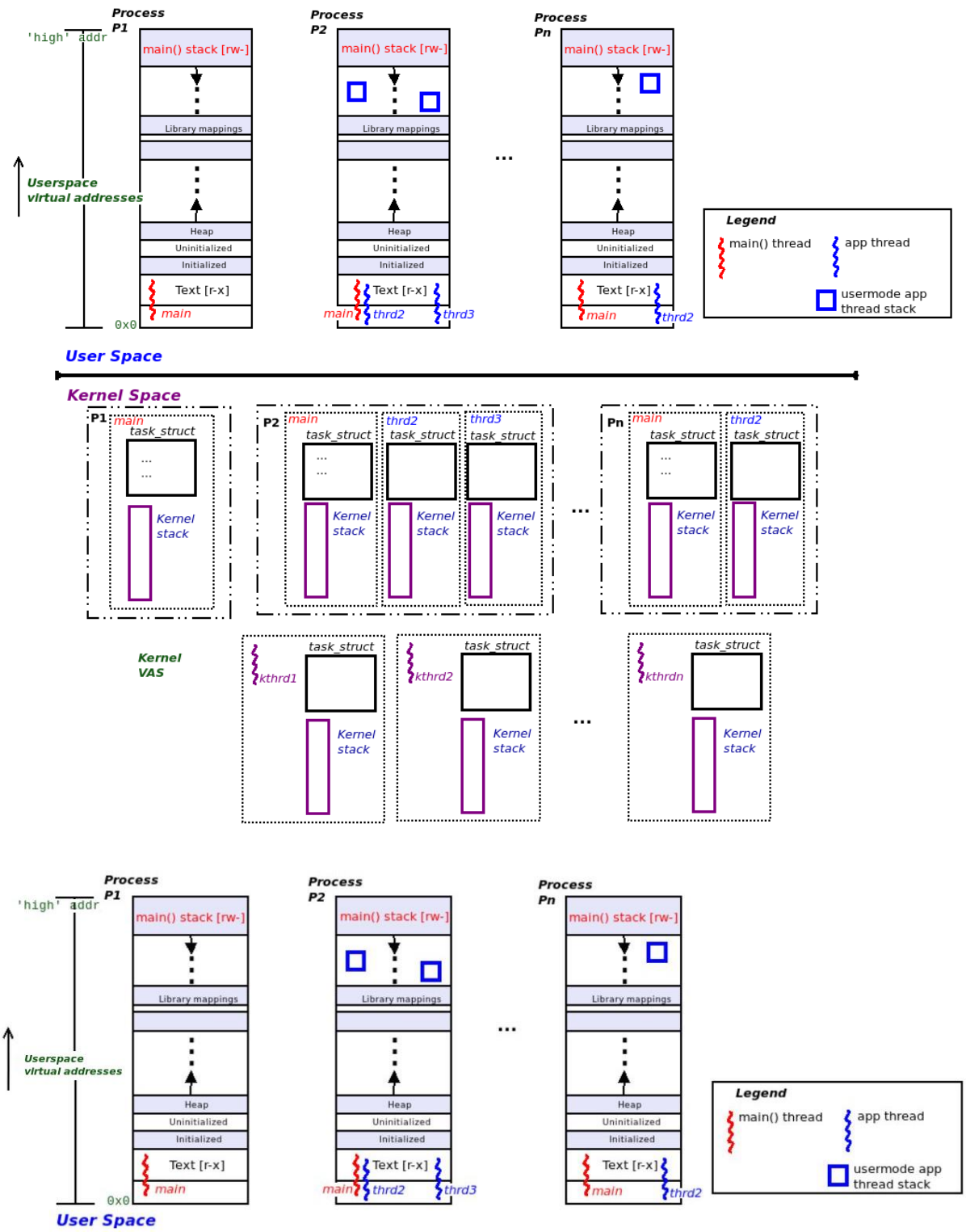
* Introduction

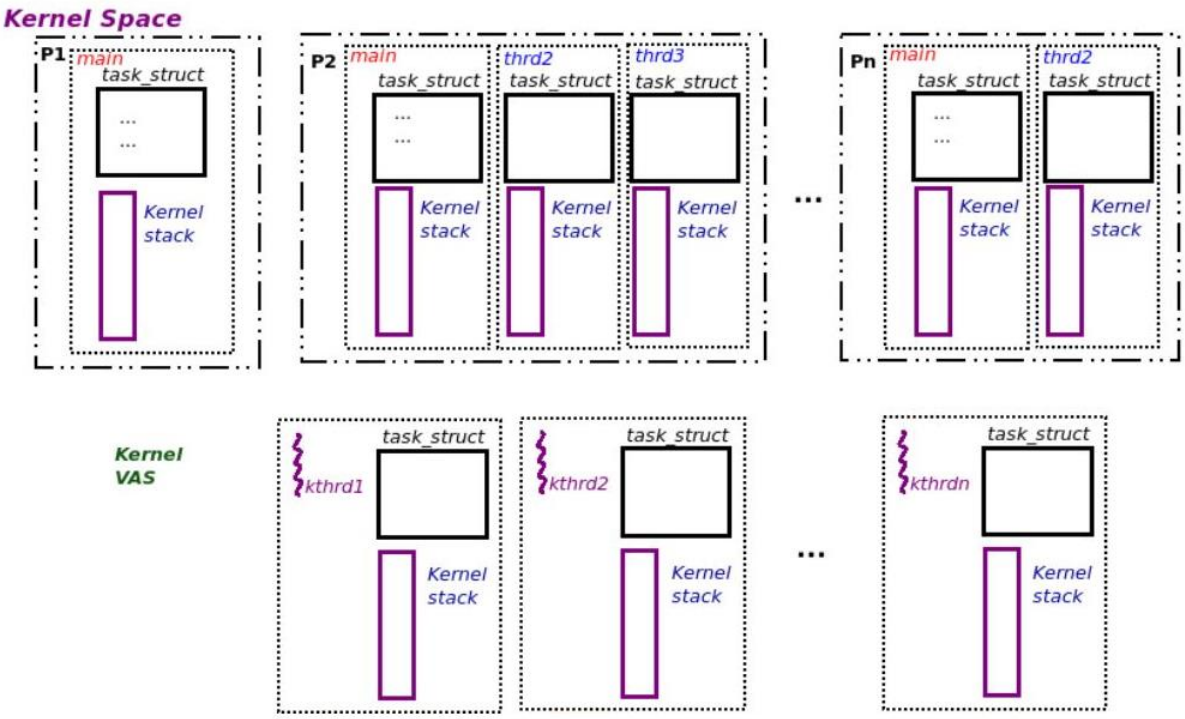
So, you want to learn how to become a Linux kernel developer? Or you have been told by your manager, "Go write a Linux driver for this device." This document's goal is to teach you everything you need to know to achieve this by describing the process you need to go through, and hints on how to work with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

Chapter 6: Kernel Internals Essentials - Processes and Threads

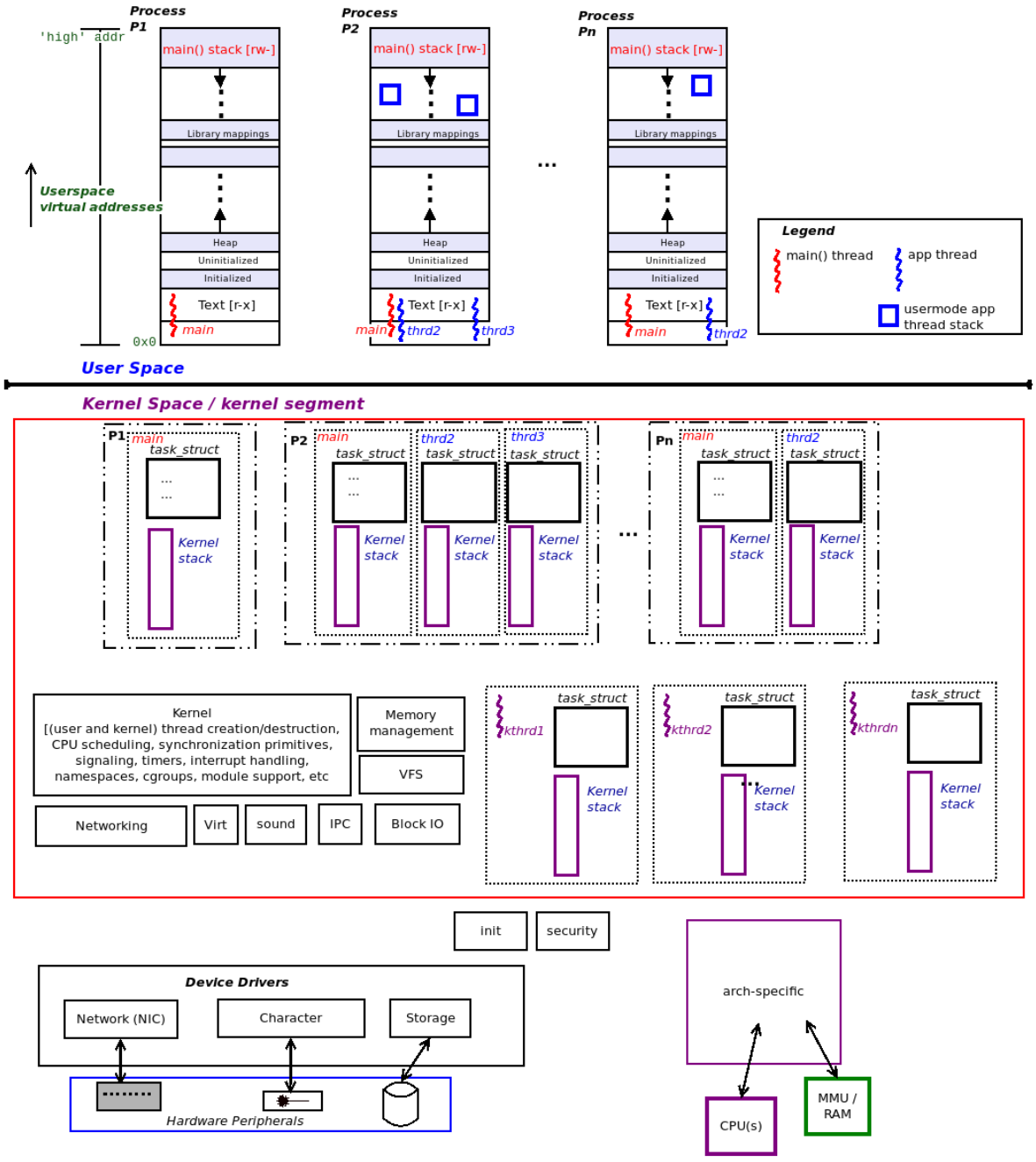




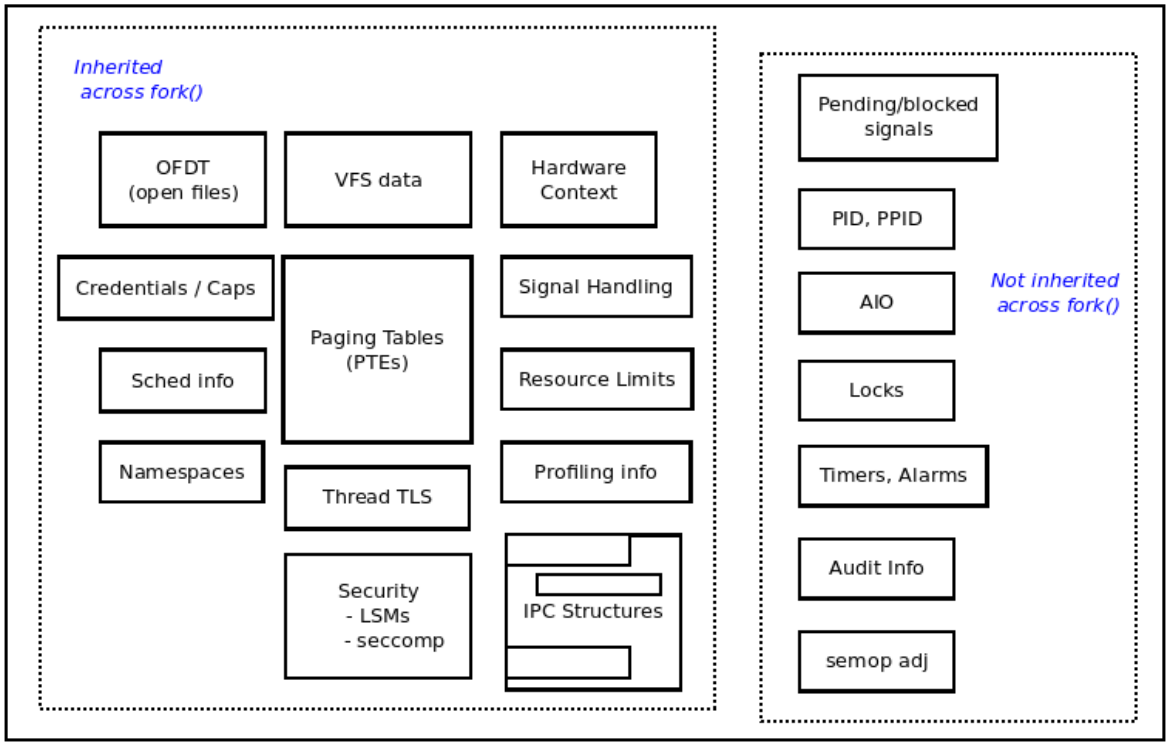

Process
P1
'high' addr
main() stack [rw-]
Library mappings
Heap
Uninitialized
Initialized
Text [r-x]
main
0x0
Userspace
virtual addresses
Process
P2
thrd2
thrd3
Process
Pn
Legend
main() thread
app thread
usermode app
thread stack
User Space
Kernel Space
task_struct
Kernel
stack
Kernel
VAS
kthrd1
kthrd2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




```
$ make
gcc -Wall -UDEBUG helloworld.c -o helloworld
strip --strip-all helloworld
gcc -g -ggdb -gdwarf-4 -O0 -Wall -Wextra -DDEBUG helloworld.c -o helloworld_dbg
$ ls
helloworld*  helloworld.c  helloworld_dbg*  Makefile  runit.sh*
$ ./runit.sh
sudo stackcount-bpfcc -p 1497640 -r .*sys_write.* -v -d
Tracing 10 functions for ".*sys_write.*"... Hit Ctrl-C to end.
^C
  ffffffffb24dde21 b'__x64_sys_write'
  ffffffffb2e0008c b'entry_SYSCALL_64_after_hwframe'
    --
  7f856dbe0057     b'[unknown]'
  49502021646c726f b'[unknown]'
    1

  ffffffffb24ddd41 b'ksys_write'
  ffffffffb22044c7 b'do_syscall_64'
  ffffffffb2e0008c b'entry_SYSCALL_64_after_hwframe'
    --
  7f856dbe0057     b'[unknown]'
  49502021646c726f b'[unknown]'
    1

Detaching...
$
```


Process P1
Process P2
Process Pn
'high' addr
0x0
main() stack [rw-]
Library mappings
Heap
Uninitialized
Initialized
Text [r-x]
main
thrd2
thrd3
Userspace virtual addresses
Legend
main() thread
app thread
usermode app thread stack
User Space
Kernel Space / kernel segment
P1
P2
Pn
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
kthrd1
kthrd2
kthrdn
init
security
Device Drivers
Network (NIC)
Character
Storage
Hardware Peripherals
arch-specific
CPU(s)
MMU / RAM

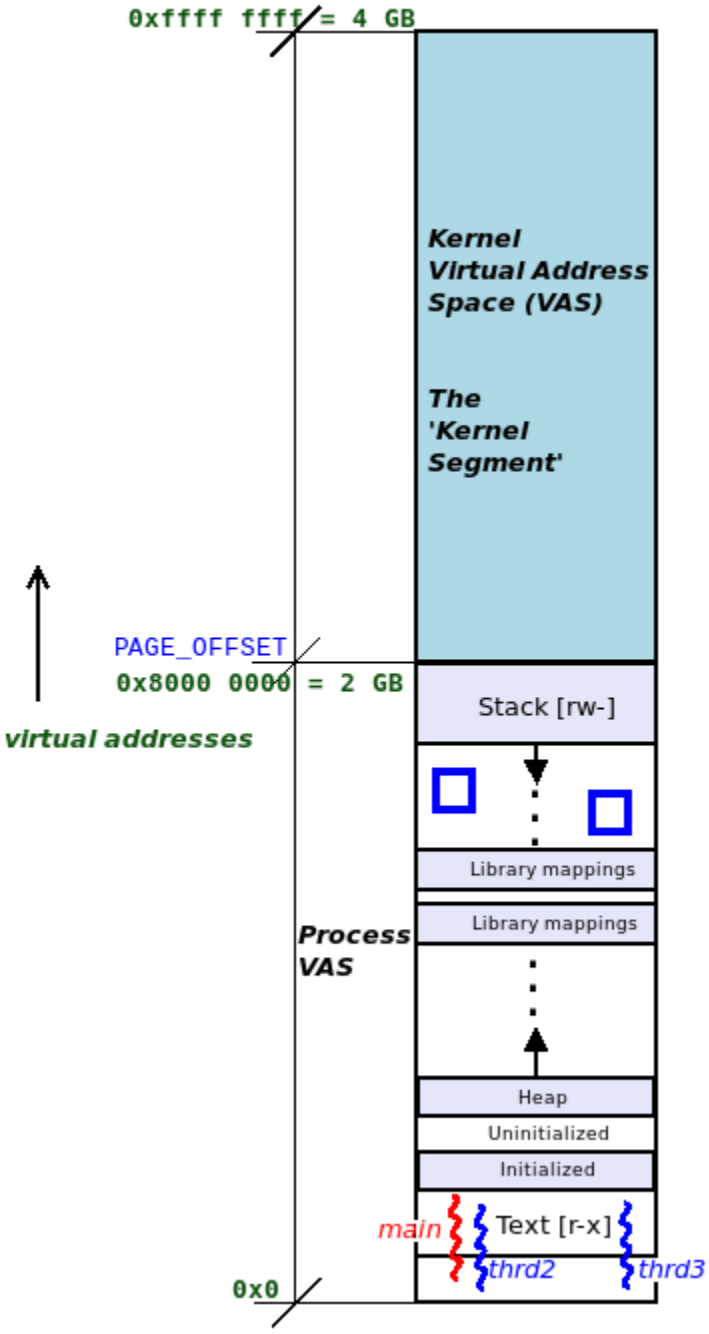
The task structure

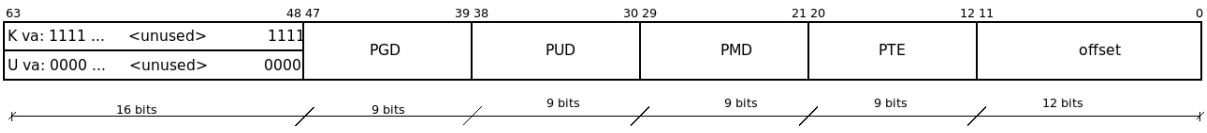


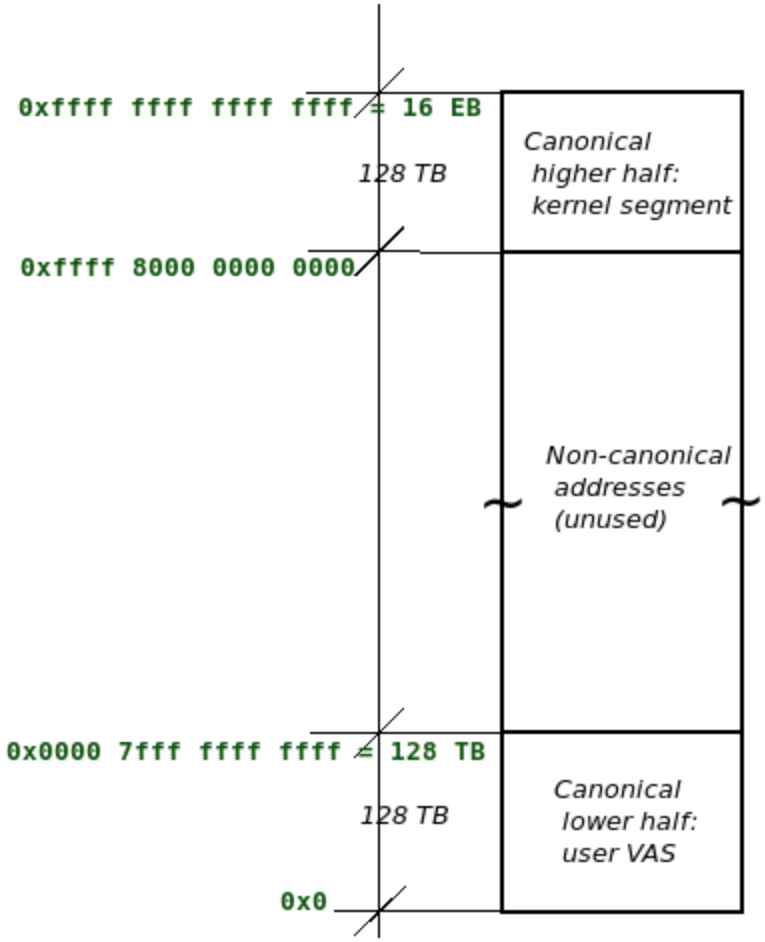
```
$ uname -r
5.4.0-llkd01
$ sudo insmod ./current_affairs.ko ; dmesg
[ 7605.102692] current_affairs: inserted
[ 7605.109628] current_affairs:show_ctx():39
[ 7605.109639] current_affairs: in process context ::
                PID         :   2205
                TGID        :   2205
                UID         :      0
                EUID        :      0 (have root)
                name        : insmod
                current (ptr to our process context's task_struct) :
                              0xffff8f4ae5d116c0 (0xffff8f4ae5d116c0)
                stack start : 0xffff9da3c16d8000 (0xffff9da3c16d8000)
$ sudo rmmod current_affairs ; dmesg | tail
[ 7616.002865] current_affairs: in process context ::
                PID         :   2209
                TGID        :   2209
                UID         :      0
                EUID        :      0 (have root)
                name        : rmmod
                current (ptr to our process context's task_struct) :
                              0xffff8f4af023ad80 (0xffff8f4af023ad80)
                stack start : 0xffff9da3c061c000 (0xffff9da3c061c000)
[ 7616.043353] current_affairs: removed
$
```

```
Threads          TGID    PID      Current            Stack-start          Thread Name   MT?#
[10287.419993]    881     881   0xffff9b09b65f8000  0xffffbaffc0998000        kerneloops
[10287.421278]    912     912   0xffff9b09e58c2d80  0xffffbaffc0a48000         VBoxClient
[10287.422776]    913     913   0xffff9b09e99e0000  0xffffbaffc0a94000         VBoxClient
[10287.424430]    938     938   0xffff9b09e99edb00  0xffffbaffc0b0c000       VBoxService    9
[10287.425889]    938     940   0xffff9b09e98496c0  0xffffbaffc0b14000          RTThrdPP
[10287.427307]    938     941   0xffff9b09fc30c440  0xffffbaffc0ad4000           control
[10287.428704]    938     942   0xffff9b09fcc596c0  0xffffbaffc0a8c000          timesync
[10287.430202]    938     943   0xffff9b09fcc5ad80  0xffffbaffc0b1c000            vminfo
[10287.431569]    938     944   0xffff9b09e99e4440  0xffffbaffc0b24000         cpuhotplug
[10287.432960]    938     945   0xffff9b09e99e16c0  0xffffbaffc0b2c000         memballoon
[10287.434417]    938     946   0xffff9b09b65fad80  0xffffbaffc0b34000            vmstats
[10287.435852]    938     947   0xffff9b09b6ae2d80  0xffffbaffc0b3c000          automount
[10287.437194]    979     979   0xffff9b09e5aadb00  0xffffbaffc06e0000               sshd
[10287.438539]    981     981   0xffff9b09e984ad80  0xffffbaffc0af4000            systemd
[10287.439832]    982     982   0xffff9b09e9848000  0xffffbaffc08f4000           (sd-pam)
[10287.441266]   1082    1082   0xffff9b09f0354440  0xffffbaffc0920000               sshd
[10287.442807]   1083    1083   0xffff9b09f03516c0  0xffffbaffc081c000               bash
[10287.444219]   1427    1427   0xffff9b09a34d0000  0xffffbaffc10dc000       packagekitd    3
[10287.445680]   1427    1428   0xffff9b09fc30ad80  0xffffbaffc10e4000             gmain
[10287.446959]   1427    1429   0xffff9b09fc3096c0  0xffffbaffc10ec000             gdbus
[10287.448340]   1748    1748   0xffff9b09b640ad80  0xffffbaffc0a08000             cupsd
[10287.449635]   1750    1750   0xffff9b09fae916c0  0xffffbaffc14f0000      cups-browsed    3
[10287.451182]   1750    1759   0xffff9b09e9b7ad80  0xffffbaffc0ae4000             gmain
[10287.452580]   1750    1760   0xffff9b09e9b7db00  0xffffbaffc1528000             gdbus
[10287.454007]   1844    1844   0xffff9b09fd5cc440  0xffffbaffc17c0000 [    kworker/u4:0]
[10287.455235]   1873    1873   0xffff9b09a34d4440  0xffffbaffc1820000 [     kworker/0:1]
[10287.456450]   1878    1878   0xffff9b09a34d16c0  0xffffbaffc1888000 [     kworker/1:1]
[10287.457668]   1879    1879   0xffff9b09a34d2d80  0xffffbaffc1810000 [    kworker/u4:2]
[10287.459160]   1882    1882   0xffff9b09a34d5b00  0xffffbaffc1768000 [    kworker/u4:1]
[10287.460920]   1887    1887   0xffff9b09fd5c96c0  0xffffbaffc18a0000               lkm
[10287.462270]   2280    2280   0xffff9b09e99ead80  0xffffbaffc1ac8000             sudo
[10287.463463]   2281    2281   0xffff9b09f0212d80  0xffffbaffc1a48000           insmod
[10287.464738] thrd_showall: total # of threads on the system: 159
```

Chapter 7: Memory Management Internals - Essentials



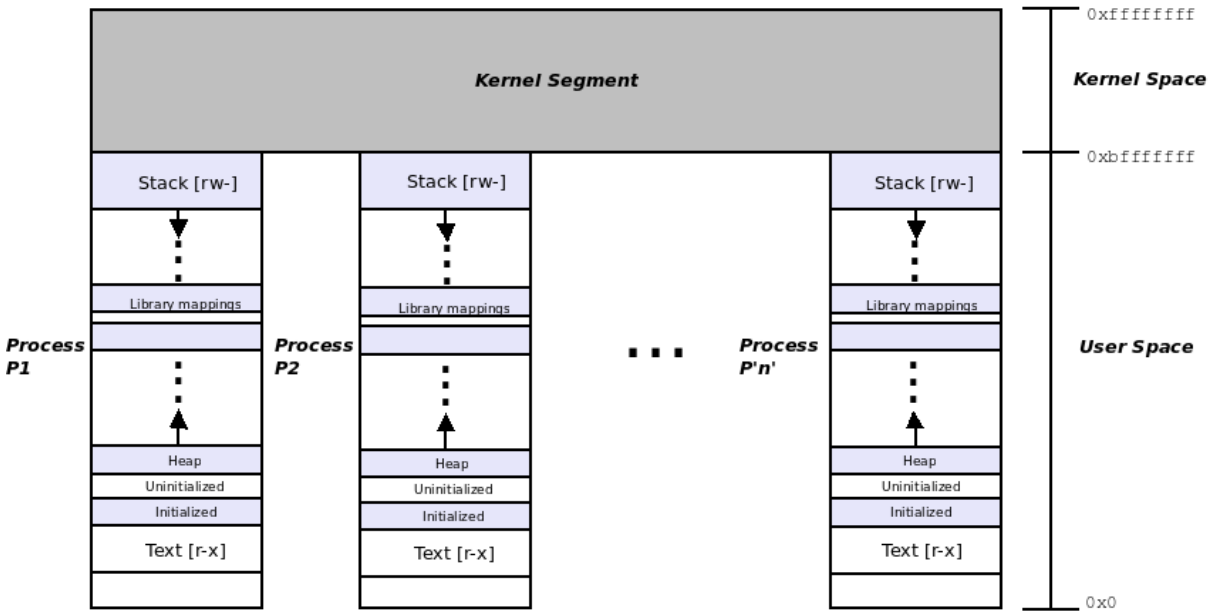




With standard 4 KB Page size

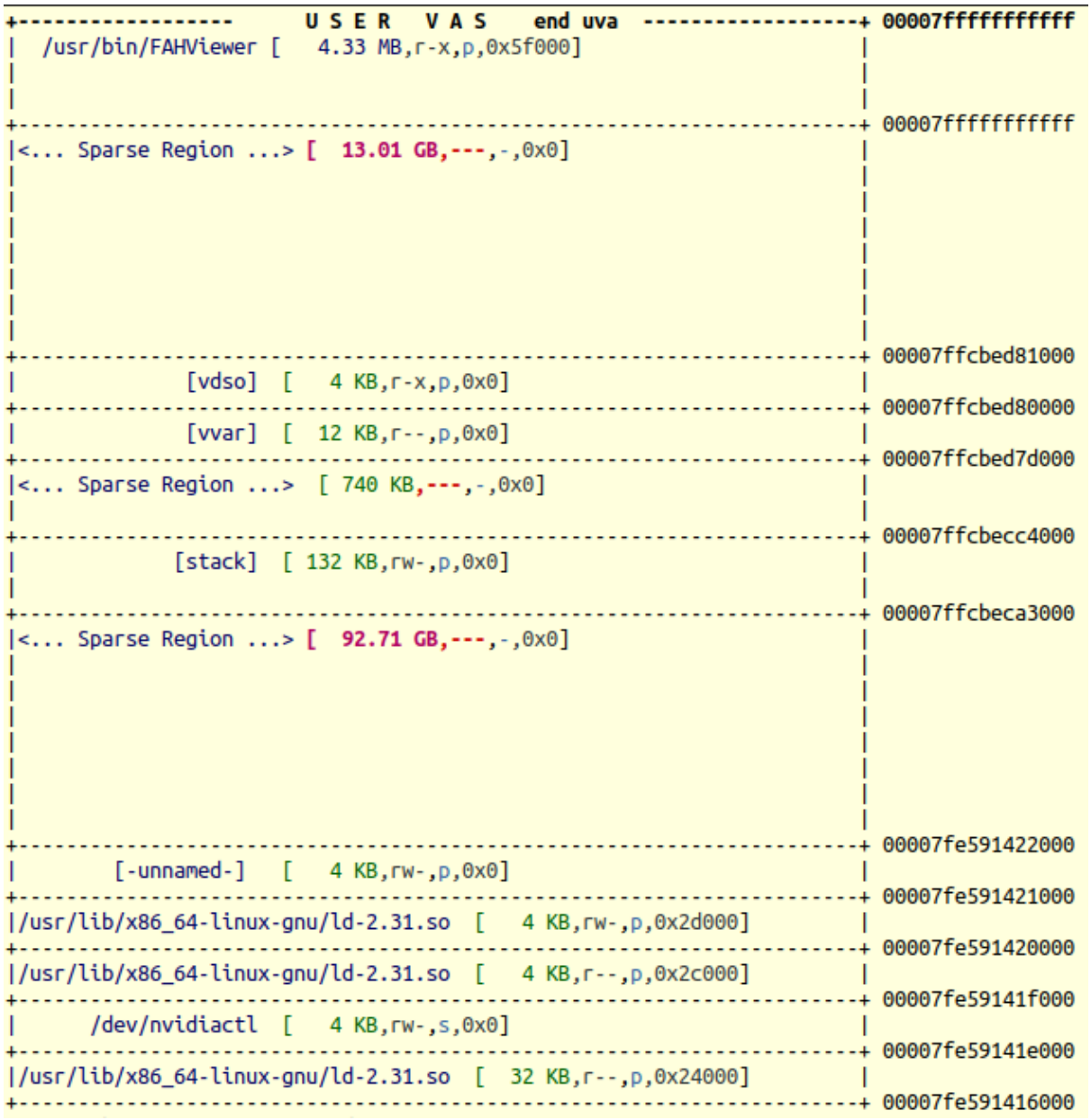
Arch	N-Level	Addr Bits	VM "Split"	Userspace		Kernel-space	
				Start vaddr	End vaddr	Start vaddr	End vaddr
IA-32	2	32	3 GB : 1 GB	0x0	0xbfff ffff	0xc000 0000	0xffff ffff
ARM	2	32	2 GB : 2 GB	0x0	0x7fff ffff	0x8000 0000	0xffff ffff
x86_64	**4**	**48**	**128 TB : 128 TB**	**0x0**	**0x0000 7fff ffff ffff**	**0xffff 8000 0000 0000**	**0xffff ffff ffff ffff**
	5*	56	64 **PB** : 64 **PB**	0x0	0x00ff ffff ffff ffff	0xff00 0000 0000 0000	0xffff ffff ffff ffff
Aarch64	3	39	512 GB : 512 GB	0x0	0x0000 007f ffff ffff	0xffff ff800 0000 000	0xffff ffff ffff ffff
	4	48	256 TB : 256 TB	0x0	0x0000 ffff ffff ffff	0xffff 0000 0000 0000	0xffff ffff ffff ffff

* *>= 4.14 Linux*

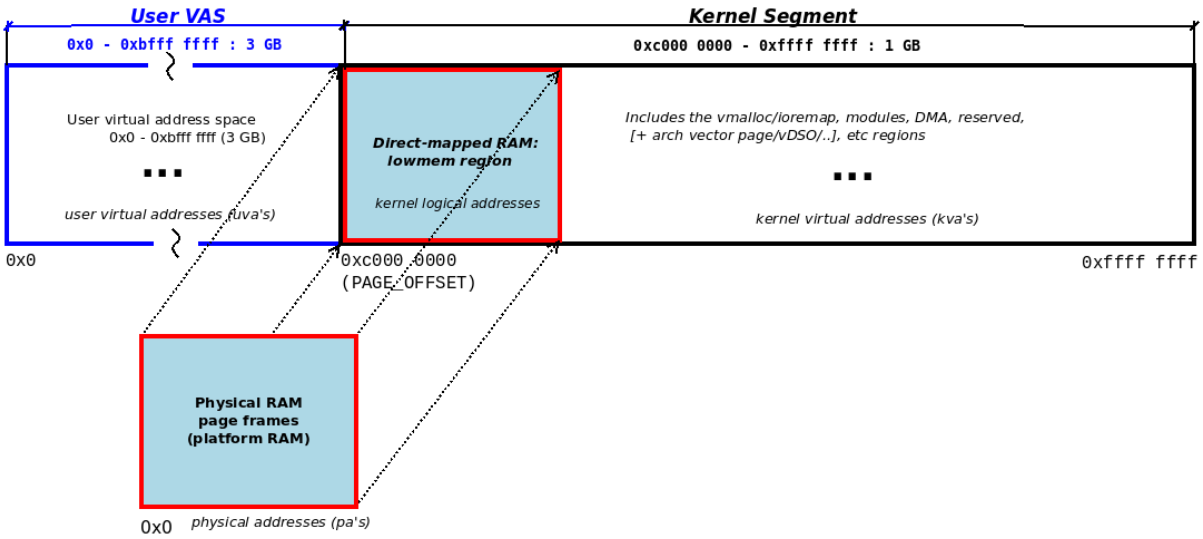



```
$ cat /proc/self/maps
555d83b65000-555d83b6d000 r-xp 00000000 08:01 524313                     /bin/cat
555d83d6c000-555d83d6d000 r--p 00007000 08:01 524313                     /bin/cat
555d83d6d000-555d83d6e000 rw-p 00008000 08:01 524313                     /bin/cat
555d840a7000-555d840c8000 rw-p 00000000 00:00 0                          [heap]
7f7d1e7e0000-7f7d1f1af000 r--p 00000000 08:01 1186501                    /usr/lib/locale/locale-archive
7f7d1f1af000-7f7d1f396000 r-xp 00000000 08:01 2102698                    /lib/x86_64-linux-gnu/libc-2.27.so
7f7d1f396000-7f7d1f596000 ---p 001e7000 08:01 2102698                    /lib/x86_64-linux-gnu/libc-2.27.so
7f7d1f596000-7f7d1f59a000 r--p 001e7000 08:01 2102698                    /lib/x86_64-linux-gnu/libc-2.27.so
7f7d1f59a000-7f7d1f59c000 rw-p 001eb000 08:01 2102698                    /lib/x86_64-linux-gnu/libc-2.27.so
7f7d1f59c000-7f7d1f5a0000 rw-p 00000000 00:00 0
7f7d1f5a0000-7f7d1f5c7000 r-xp 00000000 08:01 2102670                    /lib/x86_64-linux-gnu/ld-2.27.so
7f7d1f78c000-7f7d1f7b0000 rw-p 00000000 00:00 0
7f7d1f7c7000-7f7d1f7c8000 r--p 00027000 08:01 2102670                    /lib/x86_64-linux-gnu/ld-2.27.so
7f7d1f7c8000-7f7d1f7c9000 rw-p 00028000 08:01 2102670                    /lib/x86_64-linux-gnu/ld-2.27.so
7f7d1f7c9000-7f7d1f7ca000 rw-p 00000000 00:00 0
7fffee9ea000-7fffeea0b000 rw-p 00000000 00:00 0                          [stack]
7fffeea43000-7fffeea46000 r--p 00000000 00:00 0                          [vvar]
7fffeea46000-7fffeea48000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
$
```

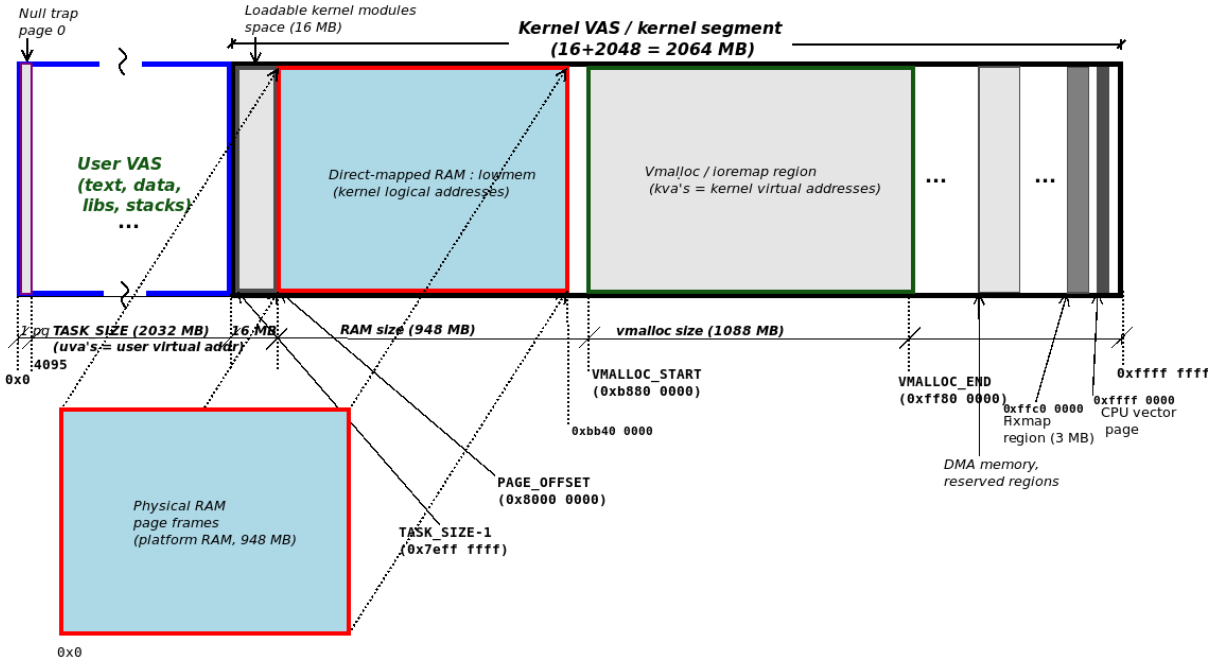
```
kaiwan $ procmap --pid=$(pgrep FAHViewer)
[i] will display memory map for process PID=6190
Detected machine type: x86_64, 64-bit system & OS

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

Sun Aug  2 14:59:40 IST 2020
[=====---  Start memory map for 6190:FAHViewer  ---=====]
[Pathname: /usr/bin/FAHViewer ]
+------------------  K E R N E L   V A S    end kva  ------------------+ ffffffffffffffff
|<... K sparse region ...> [   8.00 MB,--- ]                            |
|                                                                       |
```


U S E R V A S end uva
00007fffffffffff
/usr/bin/FAHViewer [4.33 MB,r-x,p,0x5f000]
00007fffffffffff
<... Sparse Region ...> [13.01 GB,---,-,0x0]
00007ffcbed81000
[vdso] [4 KB,r-x,p,0x0]
00007ffcbed80000
[vvar] [12 KB,r--,p,0x0]
00007ffcbed7d000
<... Sparse Region ...> [740 KB,---,-,0x0]
00007ffcbecc4000
[stack] [132 KB,rw-,p,0x0]
00007ffcbeca3000
<... Sparse Region ...> [92.71 GB,---,-,0x0]
00007fe591422000
[-unnamed-] [4 KB,rw-,p,0x0]
00007fe591421000
/usr/lib/x86_64-linux-gnu/ld-2.31.so [4 KB,rw-,p,0x2d000]
00007fe591420000
/usr/lib/x86_64-linux-gnu/ld-2.31.so [4 KB,r--,p,0x2c000]
00007fe59141f000
/dev/nvidiactl [4 KB,rw-,s,0x0]
00007fe59141e000
/usr/lib/x86_64-linux-gnu/ld-2.31.so [32 KB,r--,p,0x24000]
00007fe591416000

```
+-------------------------------------------------------------------+ 0000000000400000
|<... Sparse Region ...> [   3.99 MB,---,-,0x0]                     |
|                                                                   |
|                                                                   |
+-------------------------------------------------------------------+ 0000000001bfb000
|              [heap] [   6.46 MB,rw-,p,0x0]                        |
|                                                                   |
|                                                                   |
+-------------------------------------------------------------------+ 0000000001584000
|<... Sparse Region ...> [   2.98 MB,---,-,0x0]                     |
|                                                                   |
|                                                                   |
+-------------------------------------------------------------------+ 0000000001288000
|        [-unnamed-]   [  48 KB,rw-,p,0x0]                          |
+-------------------------------------------------------------------+ 000000000127c000
|  /usr/bin/FAHViewer  [  28 KB,rw-,p,0xe74000]                     |
+-------------------------------------------------------------------+ 0000000001275000
|  /usr/bin/FAHViewer  [ 296 KB,r--,p,0xe2a000]                     |
|                                                                   |
+-------------------------------------------------------------------+ 000000000122a000
|  /usr/bin/FAHViewer [   9.45 MB,r--,p,0x4b5000]                   |
|                                                                   |
|                                                                   |
+-------------------------------------------------------------------+ 0000000000001000
|       < NULL trap >  [   4 KB,---,-,0x0]                          |
+------------------      U S E R   V A S  start uva  ----------------+ 0000000000000000

[=====---  End memory map for 6190:FAHViewer  ---=====]
[!] stats display being skipped (see the config file)
kaiwan $
```

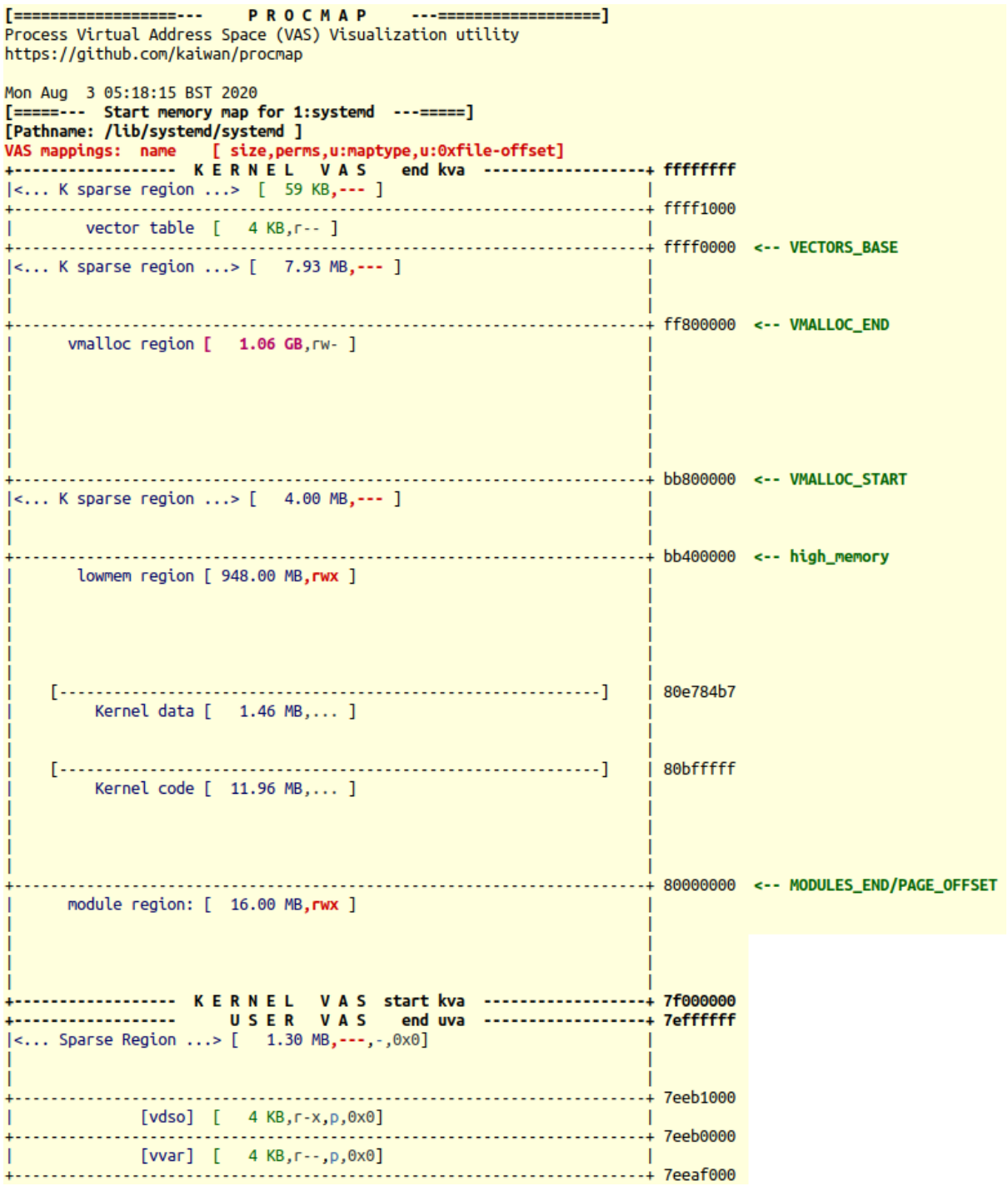


```
rpi $ sudo rmmod show_kernel_seg 2>/dev/null ; sudo dmesg -C
rpi $ sudo insmod ./show_kernel_seg.ko ; dmesg |grep -v "Journal effective setting"
[ 9930.611526] show_kernel_seg: inserted
[ 9930.617597] llkd_minsysinfo(): minimal platform info:
               CPU: ARM-32, little-endian; 32-bit OS.
[ 9930.650276]
               Some Kernel Details [by decreasing address]
               +------------------------------------------------------------+
[ 9930.650291] |vector table:        ffff0000 - ffff1000 | [   4 KB]
[ 9930.689147] |                          [ . . . ]                         |
               |fixmap region:       ffc00000 - fff00000     | [   3 MB]
[ 9930.689166] |vmalloc region:      bb800000 - ff800000     | [1088 MB =  1 GB]
[ 9930.715765] |lowmem region:       80000000 - bb400000     | [ 948 MB =  0 GB]
               |           (above:PAGE_OFFSET - highmem)                    |
[ 9930.715776] |module region:       7f000000 - 80000000 | [  16 MB]
[ 9930.715786] |                          [ . . . ]                         |
[ 9930.715800] +------------------------------------------------------------+
[ 9930.749197] show_kernel_seg: skipping show userspace...
rpi $
```



```
rpi $ ./procmap --pid=1 --verbose
[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...
 FatalError :: procmap: suitable build env for kernel modules is missing! Pl install the Linux
 kernel headers (via the appropriate package)
  Stack Call-trace:
    [frame #1] ./err_common.sh:cli_handle_error:116        <-- top of stack
    [frame #2] ./err_common.sh:FatalError:178
    [frame #3] ./lib_procmap.sh:build_lkm:209
    [frame #4] ./lib_procmap.sh:init_kernel_lkm_get_details:317
    [frame #5] ./procmap:main:0
./lib_procmap.sh: line 521: /tmp/procmap/arch_dtl: No such file or directory
```

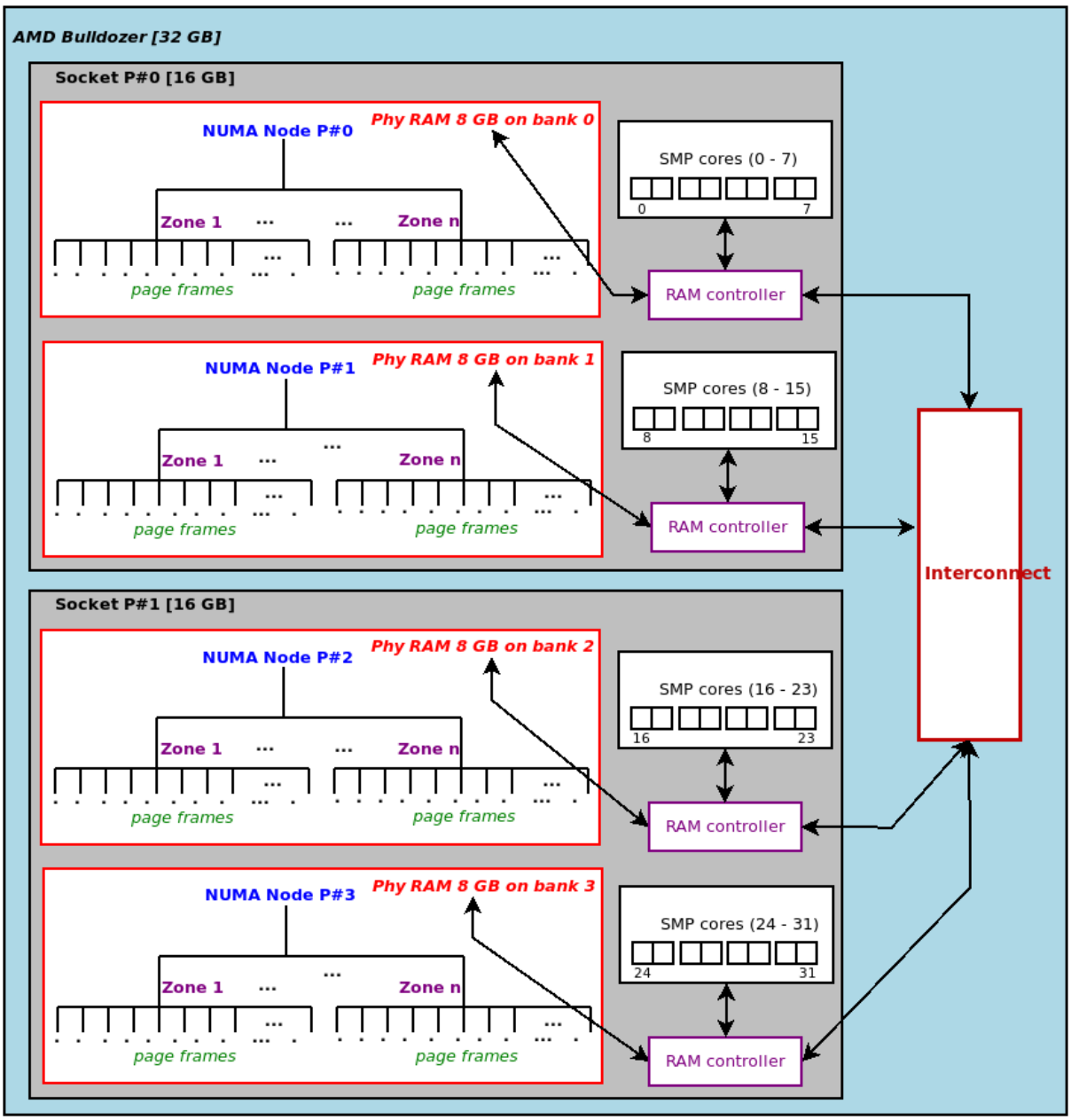
```
rpi $ ./procmap --pid=1 --verbose
[i] will display memory map for process PID=1
[i] running in VERBOSE mode
[v] kernel: init kernel LKM and get details:
[v]  debugfs location verfied
[v]  LKM inserted into kernel
[v]  debugfs file present
[v]  Parsing in various kernel variables as required
[v] set config for Aarch32:
Detected machine type: ARM-32, 32-bit OS
-------------------------------------------------
[v] System details detected ::
-------------------------------------------------
VECTORS_BASE = ffff0000
MODULES_VADDR = 7f000000
MODULES_END = 80000000
VMALLOC_START = bb800000
VMALLOC_END = ff800000
PAGE_OFFSET = 80000000
high_memory = bb400000
TASK_SIZE = 7f000000
ARCH = Aarch32
IS_64_BIT = 0
PAGE_SIZE = 4096
KERNEL_VAS_SIZE = 2164260864
USER_VAS_SIZE = 2130706432
HIGHEST_KVA = 0xffffffff
START_KVA = 7f000000
START_KVA_DEC = 2130706432
END_UVA = 7effffff
END_UVA_DEC = 2130706431
START_UVA = 0x0
-------------------------------------------------
```

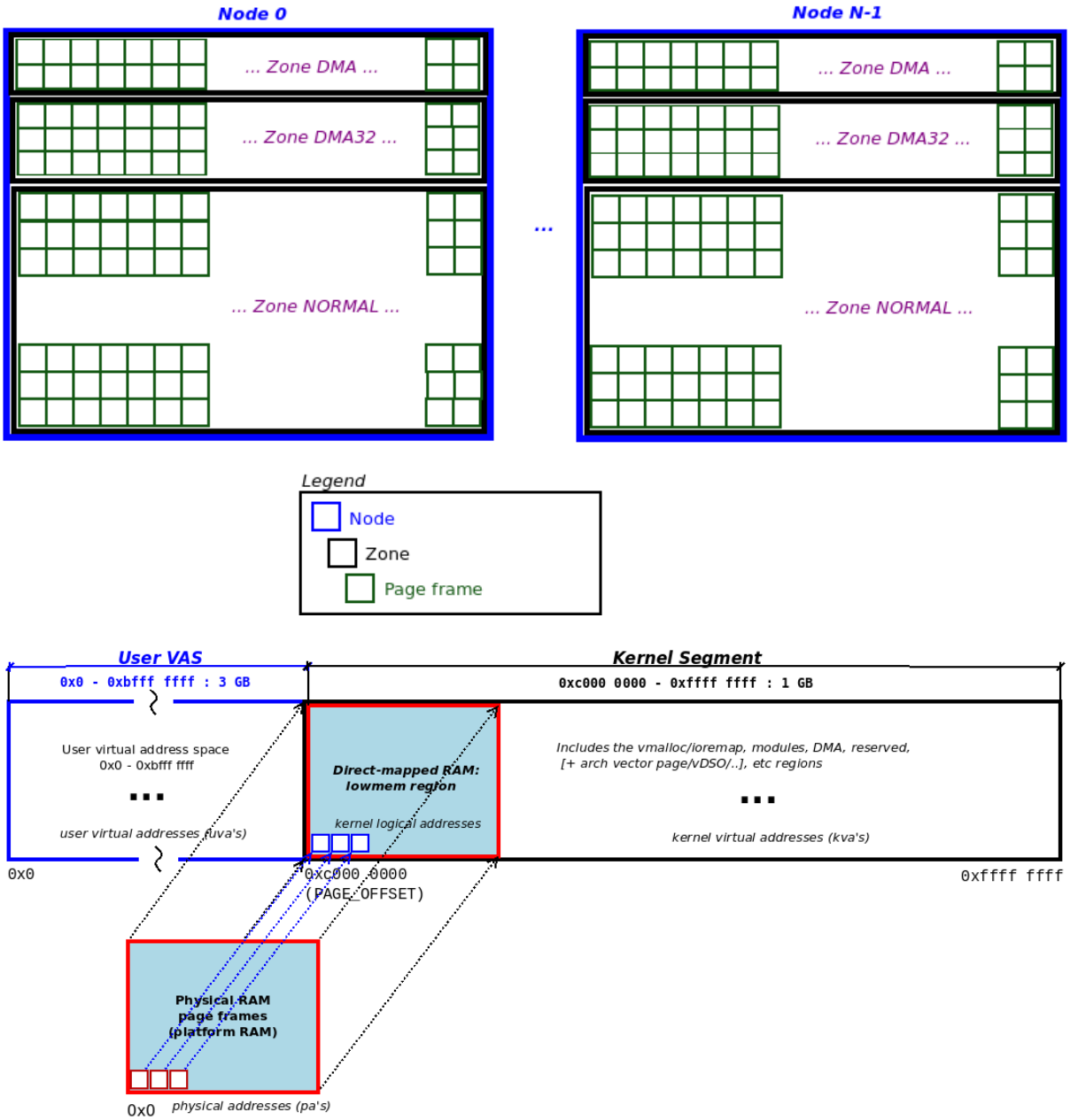

[==================---   P R O C M A P   ---==================]
Process Virtual Address Space (VAS) Visualization utility
https://github.com/kaiwan/procmap
Mon Aug 3 05:18:15 BST 2020
[=====--- Start memory map for 1:systemd ---=====]
[Pathname: /lib/systemd/systemd]
VAS mappings: name   [size,perms,u:maptype,u:0xfile-offset]
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 ...> [7.93 MB,---]
ff800000
<-- VMALLOC_END
vmalloc region [1.06 GB,rw-]
bb800000
<-- VMALLOC_START
<... K sparse region ...> [4.00 MB,---]
bb400000
<-- high_memory
lowmem region [948.00 MB,rwx]
80e784b7
Kernel data [1.46 MB,...]
80bfffff
Kernel code [11.96 MB,...]
80000000
<-- MODULES_END/PAGE_OFFSET
module region: [16.00 MB,rwx]
K E R N E L V A S start kva
7f000000
U S E R V A S end uva
7effffff
<... Sparse Region ...> [1.30 MB,---,-,0x0]
7eeb1000
[vdso] [4 KB,r-x,p,0x0]
7eeb0000
[vvar] [4 KB,r--,p,0x0]
7eeaf000

```
rpi $ uname -r
5.4.51-v7+
rpi $ sudo rmmod show_kernel_seg 2>/dev/null ; sudo dmesg -C
rpi $ sudo insmod ./show_kernel_seg.ko show_uservas=1 ; dmesg |grep -v "Journal effective setting"
[10224.062806] Voltage normalised (0x00000000)
[10235.740208] show_kernel_seg: inserted
[10235.744027] llkd_minsysinfo(): minimal platform info:
               CPU: ARM-32, little-endian; 32-bit OS.
[10235.771252]
               Some Kernel Details [by decreasing address]
               +------------------------------------------------------------+
[10235.810117] |vector table:        ffff0000 - ffff1000 | [   4 KB]
[10235.810130] |                           [ . . . ]                        |
               |fixmap region:       ffc00000 - fff00000     | [   3 MB]
[10235.810142] |vmalloc region:      bb800000 - ff800000     | [1088 MB =  1 GB]
[10235.836752] |lowmem region:       80000000 - bb400000     | [ 948 MB =  0 GB]
               |           (above:PAGE_OFFSET - highmem)                    |
[10235.836763] |module region:       7f000000 - 80000000 | [  16 MB]
[10235.870125] |                           [ . . . ]                        |
[10235.870159] +------------ Above is kernel-seg; below, user VAS  ----------+
               |                           [ . . . ]                        |
               |Process environment  7ec9a8df - 7ec9afef     | [ 1808 bytes]
               |          arguments  7ec9a8b4 - 7ec9a8df     | [   43 bytes]
               |        stack start  7ec9a7a0
               |       heap segment  01a60000 - 01a81000     | [  132 KB]
               |static data segment  0003fc48 - 00040038     | [ 1008 bytes]
               |       text segment  00010000 - 0002f430     | [  125 KB]
               |                           [ . . . ]                        |
               +------------------------------------------------------------+
[10235.909717] Above: TASK_SIZE         = 2130706432 size of userland  [  2032 MB]
                # userspace memory regions (VMAs) = 40
                Above statistics are wrt 'current' thread (see below):
[10235.909736] 003)  insmod :3989   |  .N.0   /* show_userspace_info() */
rpi $
```

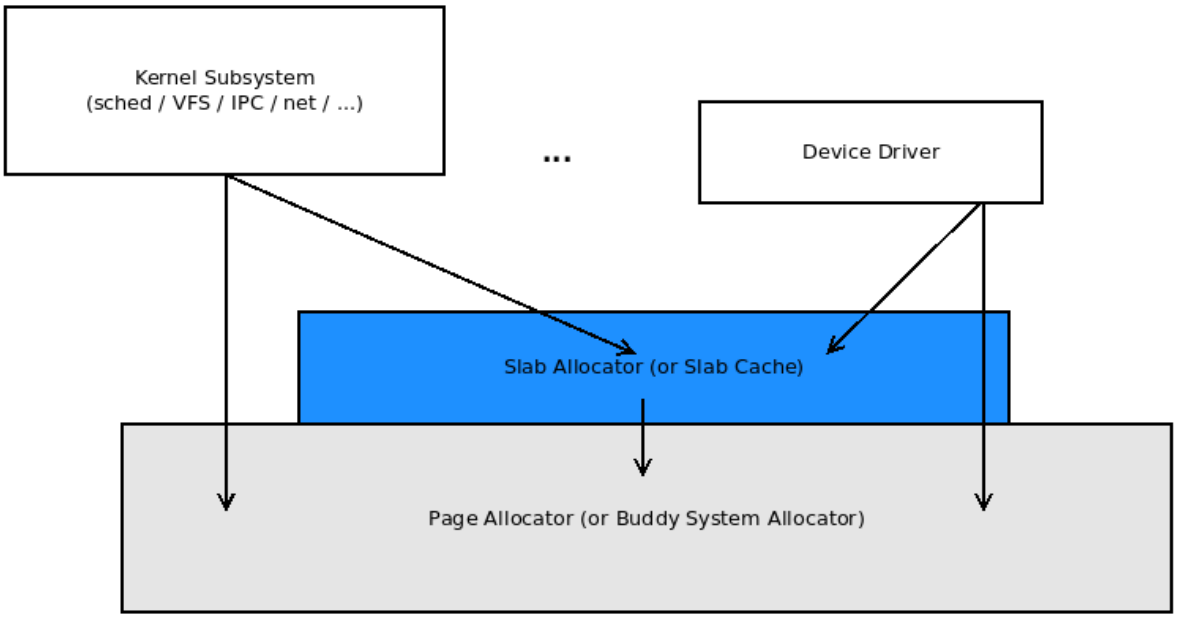
```
$ sudo ./ASLR_check.sh
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Simple [Kernel] Address Space Layout Randomization / [K]ASLR checks:
Usage: ASLR_check.sh [ASLR_value] ; where 'ASLR_value' is one of:
 0 = turn OFF ASLR
 1 = turn ON ASLR only for stack, VDSO, 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 ...
(this kernel is ver 5.4.0-llkd01, need >= 3.14)
 Kernel ASLR (KASLR) is On [default]
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
ASLR quick test:
Doing
 egrep "heap|stack" /proc/self/maps
twice:

560915f82000-560915fa3000 rw-p 00000000 00:00 0                          [heap]
7ffdb94d5000-7ffdb94f6000 rw-p 00000000 00:00 0                          [stack]

55852f9f1000-55852fa12000 rw-p 00000000 00:00 0                          [heap]
7ffc8cc04000-7ffc8cc25000 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.
```

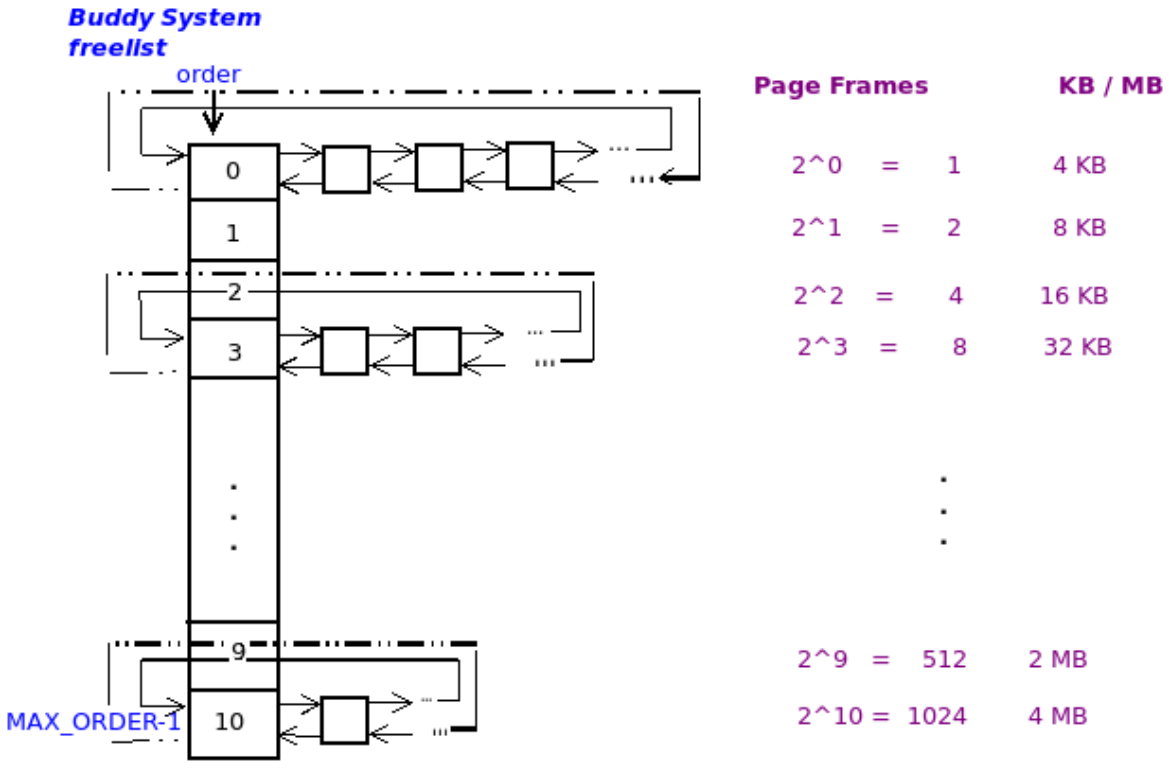
```
$ sudo ./ASLR_check.sh 0
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Simple [Kernel] Address Space Layout Randomization / [K]ASLR checks:
Usage: ASLR_check.sh [ASLR_value] ; where 'ASLR_value' is one of:
 0 = turn OFF ASLR
 1 = turn ON ASLR only for stack, 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 ...
(this kernel is ver 5.4.0-llkd01, need >= 3.14)
 Kernel ASLR (KASLR) is On [default]
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
[+] Setting (usermode) ASLR value to "0" now...
ASLR setting now is: 0
 => (usermode) ASLR is curently OFF
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
ASLR quick test:
Doing
 egrep "heap|stack" /proc/self/maps
twice:

55555578a000-5555557ac000 rw-p 00000000 00:00 0                          [heap]
7ffffffde000-7fffffffff000 rw-p 00000000 00:00 0                         [stack]

55555578a000-5555557ac000 rw-p 00000000 00:00 0                          [heap]
7ffffffde000-7fffffffff000 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)

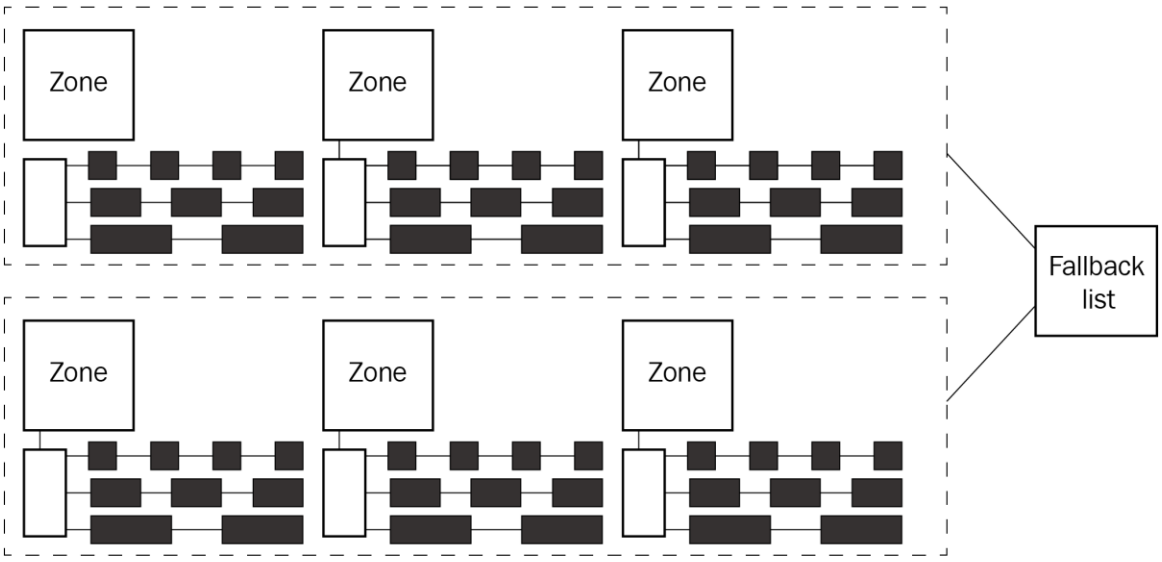
Chapter 8: Kernel Memory Allocation for Module Authors - Part 1





```
$ 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
$
```

order 0 order 1 [...] order 10



```
rpi4 $ lsmod |grep lowlevel_mem_lkm
lowlevel_mem_lkm       16384  0
rpi4 $
rpi4 $ sudo rmmod lowlevel_mem_lkm ; dmesg
[ 7769.763984] lowlevel_mem: 0. Show identity mapping: RAM page frames : kernel virtual pages :: 1:1
[ 7769.764001] show_phy_pages(): start kaddr c0000000, len 20480, contiguity_check is on
[ 7769.764012] -pg#-  ----va----   --------pa--------   -PFN-
[ 7769.764026] 00000  0xc0000000   0x0000000000000000   0
[ 7769.764039] 00001  0xc0001000   0x0000000000001000   1
[ 7769.764051] 00002  0xc0002000   0x0000000000002000   2
[ 7769.764063] 00003  0xc0003000   0x0000000000003000   3
[ 7769.764075] 00004  0xc0004000   0x0000000000004000   4
[ 7769.764093] lowlevel_mem: 1. __get_free_page() alloc'ed 1 page from the BSA @ 2b8441ff (d6350000)
[ 7769.764131] lowlevel_mem: 2. __get_free_pages() alloc'ed 2^3 = 8 page(s) = 32768 bytes
                from the BSA @ b0a14090 (d73e8000)
[ 7769.764143]  (PAGE_SIZE = 4096 bytes)
[ 7769.764155] show_phy_pages(): start kaddr d73e8000, len 32768, contiguity_check is on
[ 7769.764166] -pg#-  ----va----   --------pa--------   -PFN-
[ 7769.764178] 00000  0xd73e8000   0x00000000173e8000   95208
[ 7769.764190] 00001  0xd73e9000   0x00000000173e9000   95209
[ 7769.764202] 00002  0xd73ea000   0x00000000173ea000   95210
[ 7769.764213] 00003  0xd73eb000   0x00000000173eb000   95211
[ 7769.764225] 00004  0xd73ec000   0x00000000173ec000   95212
[ 7769.764237] 00005  0xd73ed000   0x00000000173ed000   95213
[ 7769.764249] 00006  0xd73ee000   0x00000000173ee000   95214
[ 7769.764260] 00007  0xd73ef000   0x00000000173ef000   95215
[ 7769.764278] lowlevel_mem: 3. get_zeroed_page() alloc'ed 1 page from the BSA @ a81b4775 (d63b2000)
[ 7769.764295] lowlevel_mem: 4. alloc_page() alloc'ed 1 page from the BSA @ 396e9eaf (d676e000)
                (struct page addr=026b942c (dd8364a0))
[ 7769.764313] lowlevel_mem: 5. alloc_pages() alloc'ed 32 pages from the BSA @ 83cbb79d (d6200000)
[ 7791.066874] lowlevel_mem: free-ing up the BSA memory chunks...
[ 7791.066905] lowlevel_mem: removed
rpi4 $
```

```
$ sudo rmmod lowlevel_mem_lkm 2>/dev/null ; sudo dmesg -C; sudo insmod ./lowlevel_mem_lkm.ko ; dmesg
[sudo] password for llkd:
[12747.967238] lowlevel_mem: 0. Show identity mapping: RAM page frames : kernel virtual pages :: 1:1
[12747.969619] show_phy_pages(): start kaddr ffff888000000000, len 20480, contiguity_check is on
[12747.971982] -pg#-  -------va-------     --------pa--------   --PFN--
[12747.974140] 00000  0xffff888000000000   0x0000000000000000   0
[12747.976262] 00001  0xffff888000001000   0x0000000000001000   1
[12747.978384] 00002  0xffff888000002000   0x0000000000002000   2
[12747.980340] 00003  0xffff888000003000   0x0000000000003000   3
[12747.982356] 00004  0xffff888000004000   0x0000000000004000   4
[12747.984246] lowlevel_mem: 1. __get_free_page() alloc'ed 1 page from the BSA @ ffff88804e835000 (ffff88804e835000)
[12747.988101] lowlevel_mem: 2. __get_free_pages() alloc'ed 2^3 = 8 page(s) = 32768 bytes
                from the BSA @ ffff88805d820000 (ffff88805d820000)
[12747.992492]  (PAGE_SIZE = 4096 bytes)
[12747.994432] show_phy_pages(): start kaddr ffff88805d820000, len 32768, contiguity_check is on
[12747.996710] -pg#-  -------va-------     --------pa--------   --PFN--
[12747.998893] 00000  0xffff88805d820000   0x000000005d820000   383008
[12748.001197] 00001  0xffff88805d821000   0x000000005d821000   383009
[12748.003358] 00002  0xffff88805d822000   0x000000005d822000   383010
[12748.005417] 00003  0xffff88805d823000   0x000000005d823000   383011
[12748.007451] 00004  0xffff88805d824000   0x000000005d824000   383012
[12748.009418] 00005  0xffff88805d825000   0x000000005d825000   383013
[12748.011368] 00006  0xffff88805d826000   0x000000005d826000   383014
[12748.013327] 00007  0xffff88805d827000   0x000000005d827000   383015
[12748.015712] lowlevel_mem: 3. get_zeroed_page() alloc'ed 1 page from the BSA @ ffff88804e2df000 (ffff88804e2df000)
[12748.019612] lowlevel_mem: 4. alloc_page() alloc'ed 1 page from the BSA @ ffff88804e2de000 (ffff88804e2de000)
                (struct page addr=ffffea000138b780 (ffffea000138b780))
[12748.025924] lowlevel_mem: 5. alloc_pages() alloc'ed 32 pages from the BSA @ ffff88800fe20000 (ffff88800fe20000)
$
```

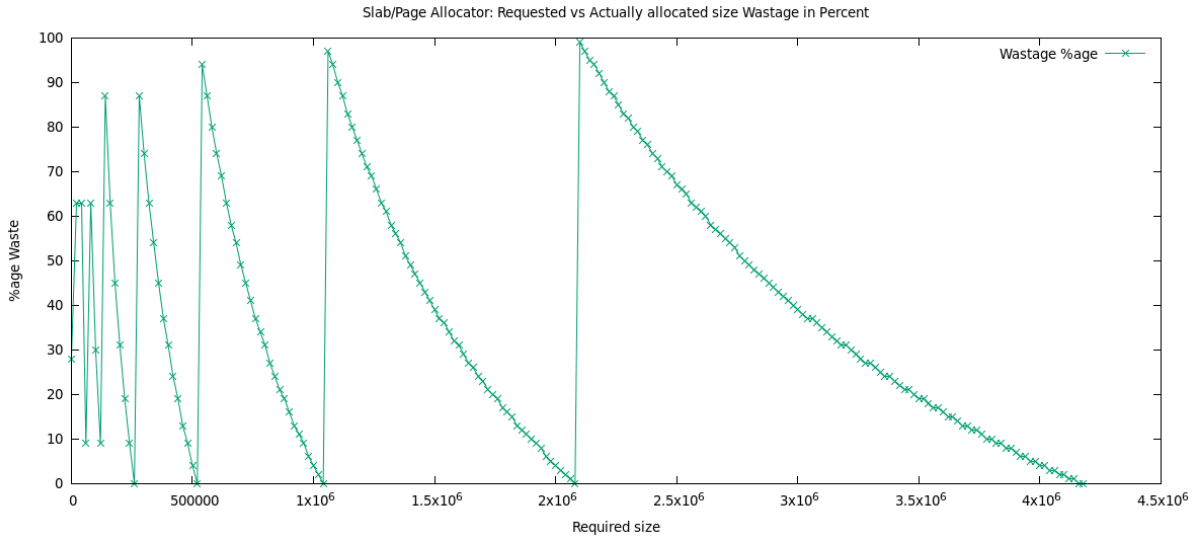
```
$ sudo vmstat -m|head -n1
Cache                       Num  Total   Size  Pages
$ sudo vmstat -m |grep --color=auto "^kmalloc"
kmalloc-8192                 45     54   8448      3
kmalloc-4096               3444   3542   4352      7
kmalloc-2048                844    975   2176     15
kmalloc-1024                891   1274   1152     14
kmalloc-512                2303   2676    640     12
kmalloc-256                1169   1248    320     12
kmalloc-192                1198   1408    256     16
kmalloc-128                1612   1617    192     21
kmalloc-96                 2080   3552    128     32
kmalloc-64                 5247   7014     96     42
kmalloc-32                 2967   3400     48     85
kmalloc-16                 5888   5888     32    128
kmalloc-8                  5333   6290     24    170
$
```

```
------------------------------
sudo insmod ./slab1.ko && lsmod|grep slab1
------------------------------
slab1                  16384  0
------------------------------
dmesg
------------------------------
[19808.873995] kmalloc() succeeds, (actual KVA) ret value = d8af6800
[19808.874005] gkptr before memset: 00000000: 00 70 af d8 00 00 00 00 00 00 00 00 00 00 00 00  .p..............
[19808.874011] gkptr before memset: 00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[19808.874017]  gkptr after memset: 00000000: 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d  mmmmmmmmmmmmmmmm
[19808.874023]  gkptr after memset: 00000010: 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d 6d  mmmmmmmmmmmmmmmm
[19808.874029] slab1: context struct alloc'ed and initialized (actual KVA ret = d8594000)
[19808.874035] ctx: 00000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[19808.874041] ctx: 00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
rpi4 slab1 $
```

```
[  391.152433] slab3_maxsize: inserted
[  391.152450] kmalloc(      0) = 0xe021e872
[  391.152466] kmalloc( 200000) = 0x018a5208
[  391.152484] kmalloc( 400000) = 0xeef720d6
[  391.152504] kmalloc( 600000) = 0xc442a50c
[  391.152519] kmalloc( 800000) = 0xc442a50c
[  391.152534] kmalloc(1000000) = 0xc442a50c
[  391.152556] kmalloc(1200000) = 0xc442a50c
[  391.152576] kmalloc(1400000) = 0xc442a50c
[  391.152597] kmalloc(1600000) = 0xc442a50c
[  391.152617] kmalloc(1800000) = 0xc442a50c
[  391.152638] kmalloc(2000000) = 0xc442a50c
[  391.152685] kmalloc(2200000) = 0x4a074daa
[  391.152720] kmalloc(2400000) = 0x4a074daa
[  391.152753] kmalloc(2600000) = 0x4a074daa
[  391.152787] kmalloc(2800000) = 0x4a074daa
[  391.152820] kmalloc(3000000) = 0x4a074daa
[  391.152853] kmalloc(3200000) = 0x4a074daa
[  391.152886] kmalloc(3400000) = 0x4a074daa
[  391.152920] kmalloc(3600000) = 0x4a074daa
[  391.152953] kmalloc(3800000) = 0x4a074daa
[  391.152987] kmalloc(4000000) = 0x4a074daa
[  391.153005] ------------[ cut here ]------------
[  391.153025] WARNING: CPU: 2 PID: 1249 at mm/page_alloc.c:4731 __alloc_pages_nodemask+0x230/0xeb8
[  391.153029] Modules linked in: slab3_maxsize(O+) rfcomm cmac bnep hci_uart btbcm bluetooth ecdh_generic ec
c 8021q garp stp llc brcmfmac brcmutil sha256_generic libsha256 cfg80211 rfkill bcm2835_codec(C) bcm2835_isp(
C) v4l2_mem2mem bcm2835_v4l2(C) raspberrypi_hwmon bcm2835_mmal_vchiq(C) videobuf2_dma_contig videobuf2_vmallo
c videobuf2_memops videobuf2_v4l2 videobuf2_common snd_bcm2835(C) videodev snd_pcm mc snd_timer vc_sm_cma(C)
snd uio_pdrv_genirq uio fixed i2c_dev ip_tables x_tables ipv6 nf_defrag_ipv6 [last unloaded: slab1]
[  391.153130] CPU: 2 PID: 1249 Comm: insmod Tainted: G         C O      5.4.51-v7+ #1
[  391.153132] Hardware name: BCM2835
[  391.153135] Backtrace:
[  391.153147] [<8010cb68>] (dump_backtrace) from [<8010ce4c>] (show_stack+0x20/0x24)
[  391.153152]  r6:b5ea2000 r5:ffffffff r4:00000000 r3:eb02066f
[  391.153161] [<8010ce2c>] (show_stack) from [<8085f21c>] (dump_stack+0xd4/0x120)
[  391.153169] [<8085f148>] (dump_stack) from [<8011e9fc>] (__warn+0xe0/0x108)
[  391.153175]  r9:0000127b r8:802ab194 r7:00000009 r6:80ab4918 r5:00000000 r4:00000000
[  391.153181] [<8011e91c>] (__warn) from [<8011eab8>] (warn_slowpath_fmt+0x94/0xa0)
[  391.153187]  r9:0000000b r8:0000127b r7:00000009 r6:80ab4918 r5:802ab194 r4:00000000
[  391.153194] [<8011ea28>] (warn_slowpath_fmt) from [<802ab194>] (__alloc_pages_nodemask+0x230/0xeb8)
[  391.153199]  r8:802c004c r7:00000cc0 r6:00401640 r5:ad800000 r4:00000000
[  391.153209] [<802aaf64>] (__alloc_pages_nodemask) from [<80288e60>] (kmalloc_order+0x2c/0x84)
[  391.153215]  r10:7f1ac088 r9:0000000b r8:802c004c r7:00000cc0 r6:00401640 r5:ad800000
```

```
[  391.153320] [<801b40c4>] (sys_finit_module) from [<80101000>] (ret_fast_syscall+0x0/0x28)
[  391.153323] Exception stack(0xb5ea3fa8 to 0xb5ea3ff0)
[  391.153328] 3fa0:                   31fa8700 7ef117c4 00000003 0002d064 00000000 00000004
[  391.153334] 3fc0: 31fa8700 7ef117c4 0003fce8 0000017b 01b237e0 00000000 00000002 00000000
[  391.153338] 3fe0: 7ef115f8 7ef115e8 00022cb8 76c46af0
[  391.153343]  r8:801011c4 r7:0000017b r6:0003fce8 r5:7ef117c4 r4:31fa8700
[  391.153347] ---[ end trace 95ab43fba62b2d3a ]---
[  391.153352] kmalloc fail, size2alloc=4200000
[  548.838970] slab3_maxsize: inserted
[  548.838988] kmalloc(      0) = 0xe021e872
[  548.839003] kmalloc( 200000) = 0xeef720d6
[  548.839020] kmalloc( 400000) = 0xeef720d6
[  548.839039] kmalloc( 600000) = 0xc442a50c
[  548.839054] kmalloc( 800000) = 0xc442a50c
[  548.839068] kmalloc(1000000) = 0xc442a50c
[  548.839091] kmalloc(1200000) = 0xc442a50c
[  548.839124] kmalloc(1400000) = 0xc442a50c
[  548.839464] kmalloc(1600000) = 0xc442a50c
[  548.839490] kmalloc(1800000) = 0xc442a50c
[  548.839510] kmalloc(2000000) = 0xc442a50c
[  548.839554] kmalloc(2200000) = 0x4a074daa
[  548.839589] kmalloc(2400000) = 0x4a074daa
[  548.839624] kmalloc(2600000) = 0x4a074daa
[  548.839658] kmalloc(2800000) = 0x4a074daa
[  548.839691] kmalloc(3000000) = 0x4a074daa
[  548.839726] kmalloc(3200000) = 0x4a074daa
[  548.839759] kmalloc(3400000) = 0x4a074daa
[  548.839793] kmalloc(3600000) = 0x4a074daa
[  548.839826] kmalloc(3800000) = 0x4a074daa
[  548.839860] kmalloc(4000000) = 0x4a074daa
[  548.839879] kmalloc fail, size2alloc=4200000
```

```
[   92.257210] slab4_actualsize: inserted
[   92.259948] kmalloc(      n) :  Actual : Wastage : Waste %
[   92.261041] kmalloc(    100) :     128 :      28 :  28%
[   92.261826] kmalloc( 200100) :  262144 :   62044 :  31%
[   92.262615] kmalloc( 400100) :  524288 :  124188 :  31%
[   92.267690] kmalloc( 600100) : 1048576 :  448476 :  74%
[   92.269786] kmalloc( 800100) : 1048576 :  248476 :  31%
[   92.271410] kmalloc(1000100) : 1048576 :   48476 :   4%
[   92.272284] kmalloc(1200100) : 2097152 :  897052 :  74%
[   92.272994] kmalloc(1400100) : 2097152 :  697052 :  49%
[   92.273695] kmalloc(1600100) : 2097152 :  497052 :  31%
[   92.274337] kmalloc(1800100) : 2097152 :  297052 :  16%
[   92.275292] kmalloc(2000100) : 2097152 :   97052 :   4%
[   92.276297] kmalloc(2200100) : 4194304 : 1994204 :  90%
[   92.277015] kmalloc(2400100) : 4194304 : 1794204 :  74%
[   92.277698] kmalloc(2600100) : 4194304 : 1594204 :  61%
[   92.278395] kmalloc(2800100) : 4194304 : 1394204 :  49%
[   92.279326] kmalloc(3000100) : 4194304 : 1194204 :  39%
[   92.280145] kmalloc(3200100) : 4194304 :  994204 :  31%
[   92.280829] kmalloc(3400100) : 4194304 :  794204 :  23%
[   92.281511] kmalloc(3600100) : 4194304 :  594204 :  16%
[   92.282192] kmalloc(3800100) : 4194304 :  394204 :  10%
[   92.282994] kmalloc(4000100) : 4194304 :  194204 :   4%
[   92.283765] ------------[ cut here ]------------
[   92.284281] WARNING: CPU: 1 PID: 1525 at mm/page_alloc.c:4738 __alloc_pages_nodemask+0x40d/0x520
[   92.285225] Modules linked in: slab4_actualsize(OE+) vboxsf(OE) vboxvideo(OE) vmwgfx snd_intel8x0
m_kms_helper snd_timer aesni_intel glue_helper syscopyarea crypto_simd snd_seq_device cryptd sysfillr
video mac_hid vboxguest(OE) sch_fq_codel parport_pc ppdev lp parport ip_tables x_tables autofs4 hid_g
[   92.290634] CPU: 1 PID: 1525 Comm: insmod Tainted: G           OE     5.4.0-llkd02-kasan #4
[   92.291549] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[   92.292446] RIP: 0010:__alloc_pages_nodemask+0x40d/0x520
[   92.294196] Code: 0f 84 2a fe ff ff 80 ce 01 e9 22 fe ff ff 4c 89 ff e8 c7 21 fa ff 49 89 c7 e9 bb
 4c 8b 3c 25 c0 fb 01
[   92.299994] RSP: 0018:ffff888035977728 EFLAGS: 00010246
```



Chapter 9: Kernel Memory Allocation for Module Authors - Part 2

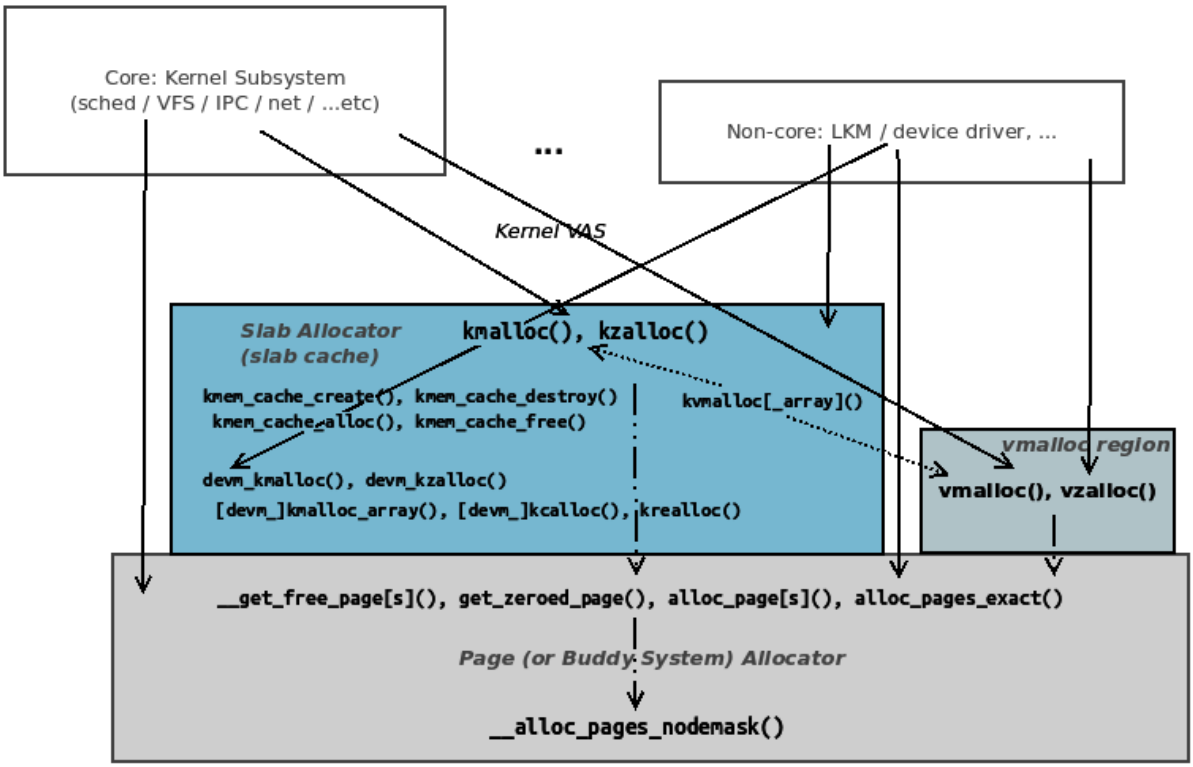
```
[25016.805844] slab_custom: inserted
[25016.809108] slab_custom: sizeof our ctx structure is 328 bytes
               using custom constructor routine? yes
[25016.816516] [ker ver > 2.6.38 cache name deprecated...]
[25016.820293] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f6440
[25016.823825] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f5e40
[25016.827274] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f4640
[25016.830510] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f6d40
[25016.833210] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f6a40
[25016.835664] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f4940
[25016.837871] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f7340
[25016.840003] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f4c40
[25016.841913] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f6740
[25016.843975] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f5b40
[25016.845800] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f5240
[25016.847559] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f4340
[25016.849319] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f7640
[25016.851086] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f4f40
[25016.852843] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f5840
[25016.854530] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f7040
[25016.856354] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f7940
[25016.858110] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f6140
[25016.859915] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f4040
[25016.861667] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f7c40
[25016.863440] slab_custom:our_ctor(): in ctor: just alloced mem object is @ 0xffff8880537f5540
[25016.865210] Our cache object (@ ffff8880537f6440, actual=ffff8880537f6440) size is 328 bytes; ksize=328
[25016.867948] obj: 00000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.870022] obj: 00000010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.872034] obj: 00000020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.873976] obj: 00000030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.875954] obj: 00000040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.877816] obj: 00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.879668] obj: 00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.881549] obj: 00000070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.883415] obj: 00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.885245] obj: 00000090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.886987] obj: 000000a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.888742] obj: 000000b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.890364] obj: 000000c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.892120] obj: 000000d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.894500] obj: 000000e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.896177] obj: 000000f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.898166] obj: 00000100: 00 00 00 00 00 00 00 00 38 38 37 35 2e 38 38 37  ........8875.887
[25016.900042] obj: 00000110: 35 2c 30 2e 33 2c 39 36 2c 30 00 00 00 00 00 00  5,0.3,96,0......
[25016.901562] obj: 00000120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.903193] obj: 00000130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[25016.904737] obj: 00000140: 00 00 00 00 00 00 00 00                          ........
```

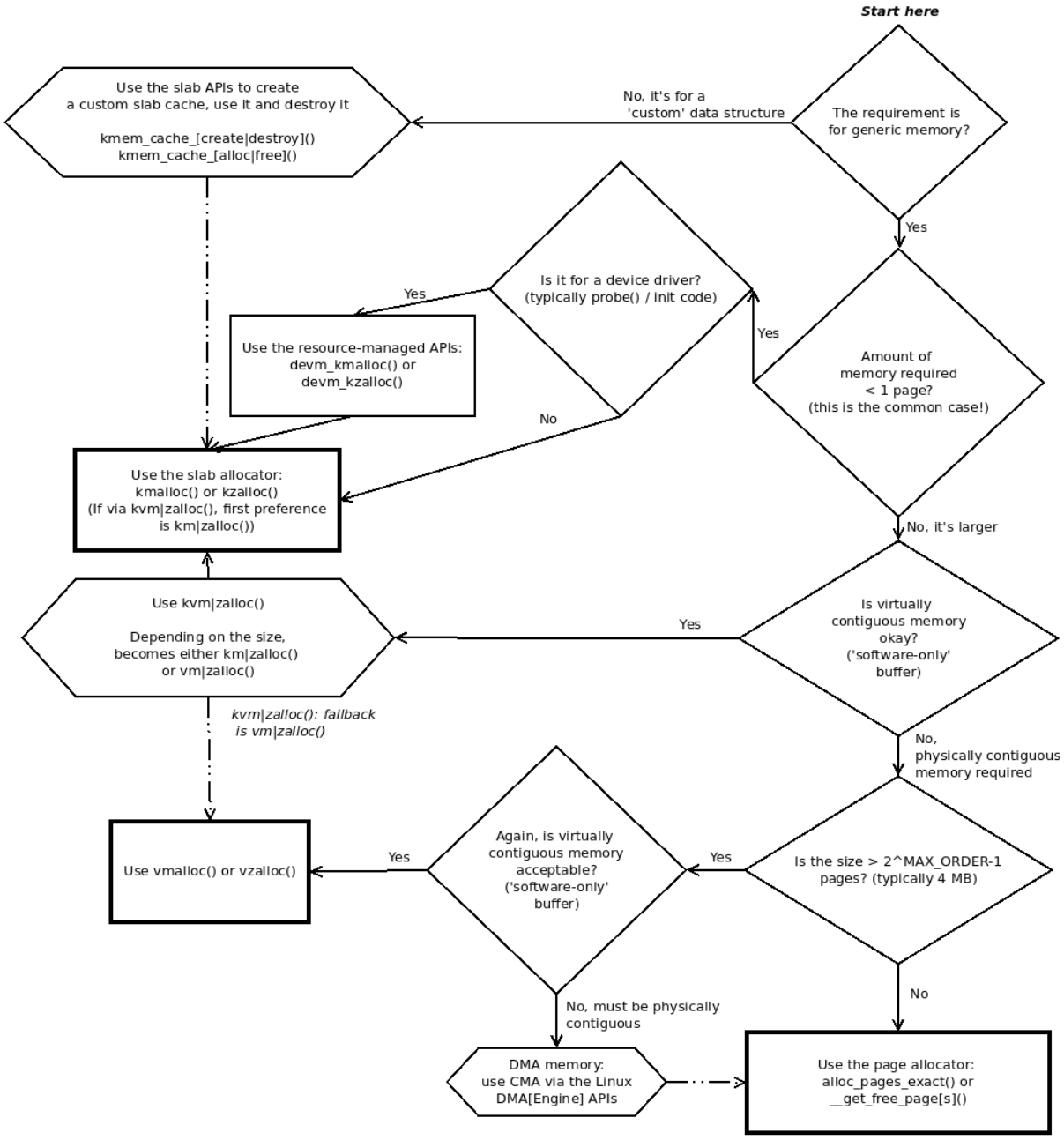
```
Jul 02 16:14:35 fed31 kernel: poison_test: custom cache destroyed; removed
Jul 02 16:16:42 fed31 kernel: poison_test: inserted
Jul 02 16:16:42 fed31 kernel: poison_test: sizeof our ctx structure is 152 bytes
                               using custom constructor routine? no
Jul 02 16:16:42 fed31 kernel: [ker ver > 2.6.38 cache name deprecated...]
Jul 02 16:16:42 fed31 kernel: Our cache object (@ 0x0000000001549e39, actual=0xffff8f7632123d80) size is 152 bytes; ksize=152
Jul 02 16:16:42 fed31 kernel: obj: 00000000: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 00000010: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 00000020: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 00000030: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 00000040: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 00000050: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 00000060: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 00000070: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 00000080: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 00000090: 6b 6b 6b 6b 6b 6b 6b 6b a5                          kkkkkkk.
Jul 02 16:16:42 fed31 kernel:  -------------- after memset s, 'z', 16 : ------------
Jul 02 16:16:42 fed31 kernel: obj: 0000000001549e39: 7a 7a 7a 7a 7a 7a 7a 7a 7a 7a 7a 7a 7a 7a 7a 7a  zzzzzzzzzzzzzzzz
Jul 02 16:16:42 fed31 kernel: obj: 00000000722b8a06: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 00000000f6326296: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 0000000068cca351: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 000000006ef6d99d: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 00000000248f0168: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 0000000048099057: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 00000000fe8d82f0: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 0000000045f90fe3: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:16:42 fed31 kernel: obj: 00000000ea67ec66: 6b 6b 6b 6b 6b 6b 6b a5                          kkkkkkk.
```

```
Jul 02 16:17:27 fed31 kernel: obj: 00000000ea67ec66: 6b 6b 6b 6b 6b 6b 6b a5                          kkkkkkk.
Jul 02 16:17:28 fed31 kernel: ============================================================================
Jul 02 16:17:28 fed31 kernel: BUG poison_test (Tainted: G    B      OE    ): Poison overwritten
Jul 02 16:17:28 fed31 kernel: ----------------------------------------------------------------------------
Jul 02 16:17:28 fed31 kernel: INFO: 0x0000000001549e39-0x00000000d178c762. First byte 0x21 instead of 0x6b
Jul 02 16:17:28 fed31 kernel: INFO: Allocated in 0xffffffffc04af0c8 age=45508 cpu=0 pid=7757
Jul 02 16:17:28 fed31 kernel:         __slab_alloc+0x1c/0x30
Jul 02 16:17:28 fed31 kernel:         kmem_cache_alloc+0x23e/0x270
Jul 02 16:17:28 fed31 kernel:         0xffffffffc04af0c8
Jul 02 16:17:28 fed31 kernel:         do_one_initcall+0x6e/0x254
Jul 02 16:17:28 fed31 kernel:         do_init_module+0x5c/0x230
Jul 02 16:17:28 fed31 kernel:         load_module+0x2758/0x2a20
Jul 02 16:17:28 fed31 kernel:         __do_sys_finit_module+0xaa/0x110
Jul 02 16:17:28 fed31 kernel:         do_syscall_64+0x5b/0x180
Jul 02 16:17:28 fed31 kernel:         entry_SYSCALL_64_after_hwframe+0x44/0xa9
Jul 02 16:17:28 fed31 kernel: INFO: Freed in slab_custom_exit+0x13/0xf2d [poison_test] age=15 cpu=0 pid=7785
Jul 02 16:17:28 fed31 kernel:         kmem_cache_free+0x2df/0x300
Jul 02 16:17:28 fed31 kernel:         slab_custom_exit+0x13/0xf2d [poison_test]
Jul 02 16:17:28 fed31 kernel:         __x64_sys_delete_module+0x13f/0x280
Jul 02 16:17:28 fed31 kernel:         do_syscall_64+0x5b/0x180
Jul 02 16:17:28 fed31 kernel:         entry_SYSCALL_64_after_hwframe+0x44/0xa9
Jul 02 16:17:28 fed31 kernel: INFO: Slab 0x000000002a6b69d9 objects=14 used=0 fp=0x0000000001549e39 flags=0xfffe000010200
Jul 02 16:17:28 fed31 kernel: INFO: Object 0x0000000001549e39 @offset=7552 fp=0x000000007b344c6b
Jul 02 16:17:28 fed31 kernel: Redzone 000000003e2471ad: bb bb bb bb bb bb bb bb bb bb bb bb bb bb bb bb  ................
Jul 02 16:17:28 fed31 kernel: Redzone 00000000406be0d4: bb bb bb bb bb bb bb bb bb bb bb bb bb bb bb bb  ................
Jul 02 16:17:28 fed31 kernel: Redzone 000000001badcd95: bb bb bb bb bb bb bb bb bb bb bb bb bb bb bb bb  ................
Jul 02 16:17:28 fed31 kernel: Redzone 00000000475f60c2: bb bb bb bb bb bb bb bb bb bb bb bb bb bb bb bb  ................
Jul 02 16:17:28 fed31 kernel: Object 0000000001549e39: 21 21 21 21 21 21 21 21 21 6b 6b 6b 6b 6b 6b 6b  !!!!!!!!!kkkkkkk
Jul 02 16:17:28 fed31 kernel: Object 00000000722b8a06: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:17:28 fed31 kernel: Object 00000000f6326296: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:17:28 fed31 kernel: Object 0000000068cca351: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:17:28 fed31 kernel: Object 000000006ef6d99d: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:17:28 fed31 kernel: Object 00000000248f0168: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:17:28 fed31 kernel: Object 0000000048099057: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:17:28 fed31 kernel: Object 00000000fe8d82f0: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:17:28 fed31 kernel: Object 0000000045f90fe3: 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b 6b  kkkkkkkkkkkkkkkk
Jul 02 16:17:28 fed31 kernel: Object 00000000ea67ec66: 6b 6b 6b 6b 6b 6b 6b a5                          kkkkkkk.
Jul 02 16:17:28 fed31 kernel: Redzone 0000000008937cab7: bb bb bb bb bb bb bb bb                        ........
Jul 02 16:17:28 fed31 kernel: Padding 00000000c1e31d5b: 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a  ZZZZZZZZZZZZZZZZ
Jul 02 16:17:28 fed31 kernel: Padding 00000000d1db10a0: 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a 5a  ZZZZZZZZZZZZZZZZ
Jul 02 16:17:28 fed31 kernel: Padding 0000000001d18bd2: 5a 5a 5a 5a 5a 5a 5a 5a                          ZZZZZZZZ
Jul 02 16:17:28 fed31 kernel: CPU: 0 PID: 7785 Comm: rmmod Tainted: G    B      OE     5.4.0-llkd01 #2
Jul 02 16:17:28 fed31 kernel: Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
Jul 02 16:17:28 fed31 kernel: Call Trace:
Jul 02 16:17:28 fed31 kernel:  dump_stack+0x66/0x90
Jul 02 16:17:28 fed31 kernel:  check_bytes_and_report.cold+0x40/0x58
Jul 02 16:17:28 fed31 kernel:  check_object+0x20d/0x250
Jul 02 16:17:28 fed31 kernel:  __free_slab+0x9e/0x380
```

```
[   65.792406] vmalloc_demo: loading out-of-tree module taints kernel.
[   65.792439] vmalloc_demo: module verification failed: signature and/or required key missing
 - tainting kernel
[   65.792943] vmalloc_demo: inserted
[   65.792949] 1. vmalloc():   vptr_rndm = 0x00000000fcc77e4d (actual=0xffffa858c080d000)
[   65.792951]  content: 4f 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  O...............
[   65.792955] 2. vzalloc():   vptr_init = 0x00000000c35b38e5 (actual=0xffffa858c0821000)
[   65.792956]  content: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[   65.793562] 3. kvmalloc() :        kv = 0x00000000fb2af97f (actual=0xffffa858c2c09000)
                   (for 5242880 bytes)
[   65.793564]  content: ca ef 00 00 00 00 00 cc 1a 01 00 00 00 00 00 00  ................
[   65.793573] 4. kcalloc() :      kvarr = 0x00000000d0418057 (actual=0xffff89f97b49a000)
[   65.793574]  content: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[   65.793596] 5. __vmalloc():       vrx = 0x00000000d4d28888 (actual=0xffffa858c1971000)
[   65.793597]  content: 75 70 00 2e 61 6e 6e 6f 62 69 6e 5f 67 72 6f 75  up..annobin_grou
```

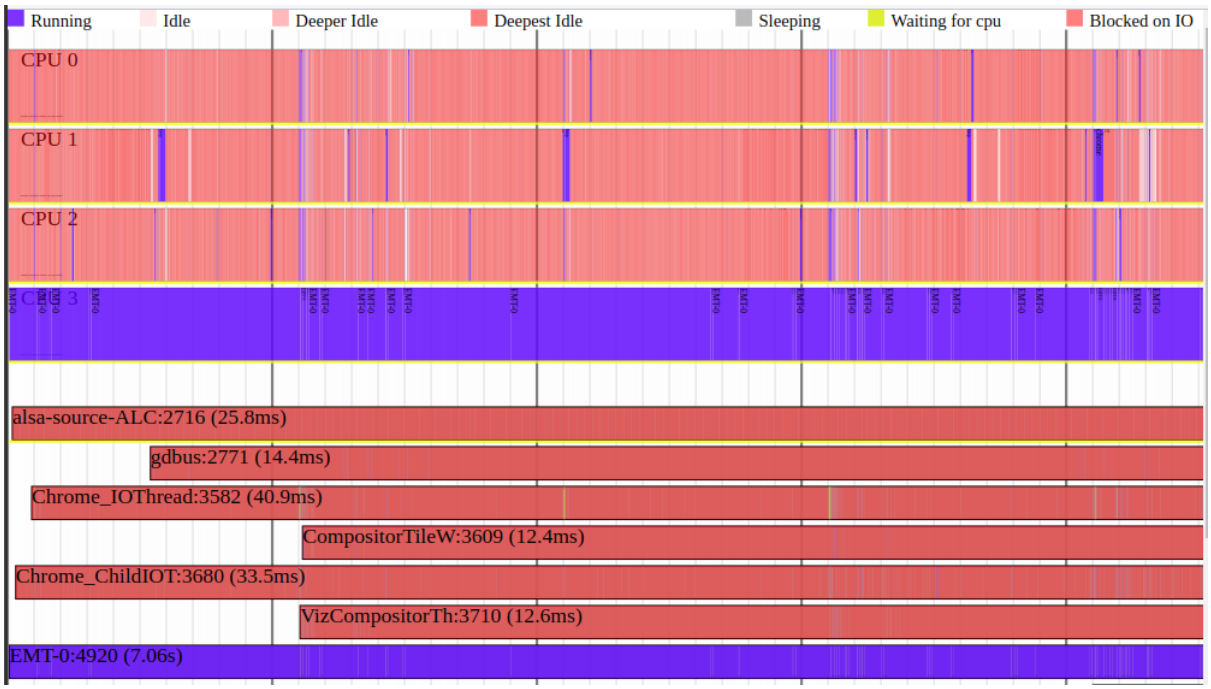
```
[ 1199.357144] vmalloc_demo: inserted
[ 1199.357154] 1. vmalloc():   vptr_rndm = 0x00000000203f6102 (actual=0xffffa858c016d000)
[ 1199.357156]  content: dd 03 00 00 00 00 00 b2 00 00 00 00 00 00 00 00  ................
[ 1199.357163] 2. vzalloc():   vptr_init = 0x000000001f29018a (actual=0xffffa858c0197000)
[ 1199.357165]  content: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[ 1199.358586] 3. kvmalloc() :        kv = 0x000000007c676ba4 (actual=0xffffa858c2ba9000)
                   (for 5242880 bytes)
[ 1199.358589]  content: 63 cd 00 00 00 00 00 e4 1a 01 00 00 00 00 00 00  c...............
[ 1199.358591] 4. kcalloc() :      kvarr = 0x000000002829c3ec (actual=0xffff89f97bcee000)
[ 1199.358593]  content: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
[ 1199.358609] 5. __vmalloc():       vrx = 0x000000008dd6a024 (actual=0xffffa858c1a39000)
[ 1199.358610]  content: 55 16 1f b7 e3 b8 e6 04 00 00 00 00 00 00 00 00  U...............
[ 1199.358615] BUG: unable to handle page fault for address: ffffa858c1a39004
[ 1199.358726] #PF: supervisor write access in kernel mode
[ 1199.358727] #PF: error_code(0x0003) - permissions violation
[ 1199.358729] PGD 7d544067 P4D 7d544067 PUD 7d545067 PMD 341d1067 PTE 8000000075bd6061
[ 1199.358735] Oops: 0003 [#1] SMP PTI
[ 1199.358739] CPU: 1 PID: 3012 Comm: insmod Tainted: G           OE     5.4.0-llkd01 #2
[ 1199.358740] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[ 1199.358745] RIP: 0010:vmalloc_demo_init+0x2c9/0x1000 [vmalloc_demo]
[ 1199.358747] Code: d2 b9 10 00 00 00 6a 10 41 b8 01 00 00 00 48 c7 c6 83 c0 49 c0 48 c7 c7 8e c0
 49 c0 e8 f0 07 05 c3 58 5a 48 8b 05 77 d2 ff ff <48> c7 40 04 ba 00 00 00 31 c0 c3 48 8b 3d 75 d2
 ff ff e8 20 13 e1
[ 1199.358749] RSP: 0018:ffffa858c09c7c78 EFLAGS: 00010286
[ 1199.358751] RAX: ffffa858c1a39000 RBX: 0000000000000000 RCX: 0000000000000006
[ 1199.358753] RDX: 0000000000000001 RSI: ffffffff8445c358 RDI: ffff89f97db17c80
[ 1199.358754] RBP: ffffffffc04a1000 R08: 0000000000000000 R09: 0000000000000000
[ 1199.358756] R10: 0000000000000001 R11: 0000000000000001 R12: ffff89f97842a720
[ 1199.358757] R13: ffff89f933107830 R14: ffffffffc049e140 R15: ffffffffc049e190
[ 1199.358759] FS:  00007ff3faf3d740(0000) GS:ffff89f97db00000(0000) knlGS:0000000000000000
[ 1199.358761] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[ 1199.358762] CR2: ffffa858c1a39004 CR3: 0000000072a7a005 CR4: 00000000000606e0
[ 1199.358767] Call Trace:
[ 1199.358772]  do_one_initcall+0x6e/0x254
[ 1199.358789]  ? _cond_resched+0x15/0x30
[ 1199.358792]  ? kmem_cache_alloc_trace+0x1da/0x280
[ 1199.358797]  do_init_module+0x5c/0x230
[ 1199.358804]  load_module+0x2758/0x2a20
[ 1199.358810]  ? vfs_read+0x148/0x170
[ 1199.358816]  ? __do_sys_finit_module+0xaa/0x110
[ 1199.358818]  __do_sys_finit_module+0xaa/0x110
[ 1199.358824]  do_syscall_64+0x5b/0x180
[ 1199.358827]  entry_SYSCALL_64_after_hwframe+0x44/0xa9
[ 1199.358832] RIP: 0033:0x7ff3fb06715d
```

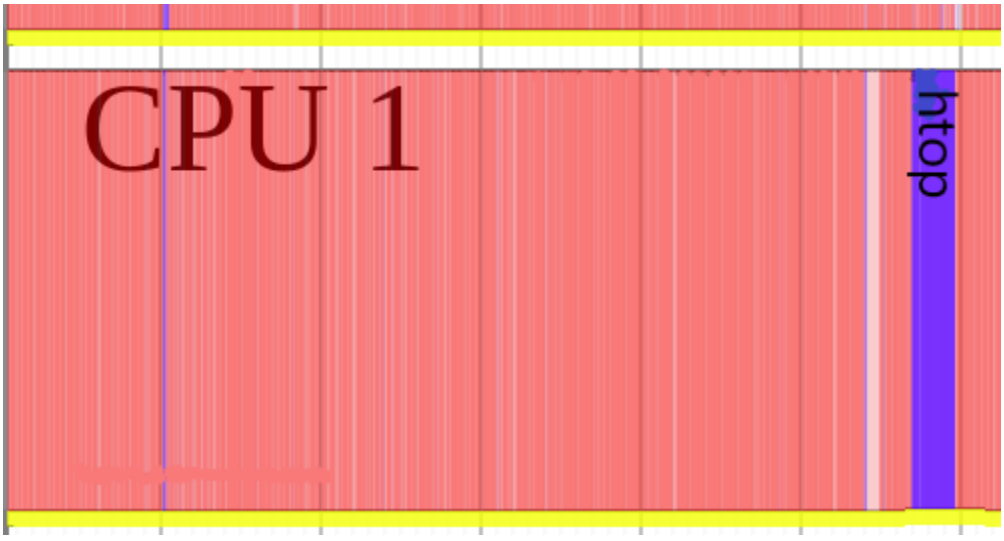

Core: Kernel Subsystem
(sched / VFS / IPC / net / ...etc)
...
Non-core: LKM / device driver, ...
Kernel VAS
Slab Allocator
(slab cache)
kmalloc(), kzalloc()
kmem_cache_create(), kmem_cache_destroy()
kmem_cache_alloc(), kmem_cache_free()
kvmalloc[_array]()
devm_kmalloc(), devm_kzalloc()
[devm_]kmalloc_array(), [devm_]kcalloc(), krealloc()
vmalloc region
vmalloc(), vzalloc()
__get_free_page[s](), get_zeroed_page(), alloc_page[s](), alloc_pages_exact()
Page (or Buddy System) Allocator
__alloc_pages_nodemask()

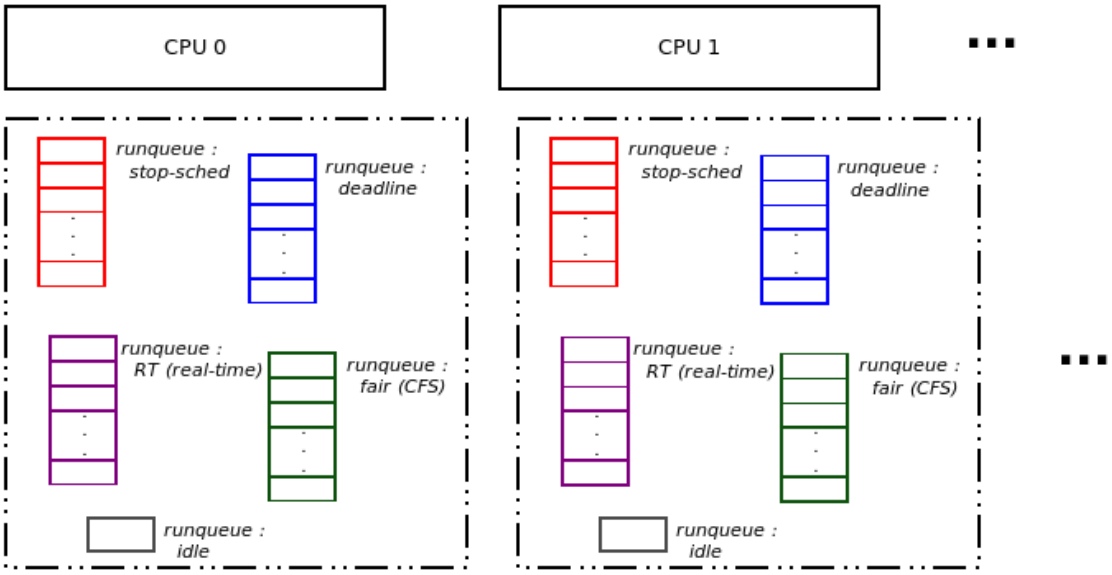

Start here
The requirement is for generic memory?
No, it's for a 'custom' data structure
Use the slab APIs to create a custom slab cache, use it and destroy it
kmem_cache_[create|destroy]()
kmem_cache_[alloc|free]()
Yes
Amount of memory required < 1 page? (this is the common case!)
Yes
Is it for a device 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 kvm|zalloc(), first preference is km|zalloc())
No, it's larger
Is virtually contiguous memory okay? ('software-only' buffer)
Yes
Use kvm|zalloc()
Depending on the size, becomes either km|zalloc() or vm|zalloc()
kvm|zalloc(): fallback is vm|zalloc()
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() or vzalloc()
No
Use the page allocator: alloc_pages_exact() or __get_free_page[s]()
No, must be physically contiguous
DMA memory: use CMA via the Linux DMA[Engine] APIs

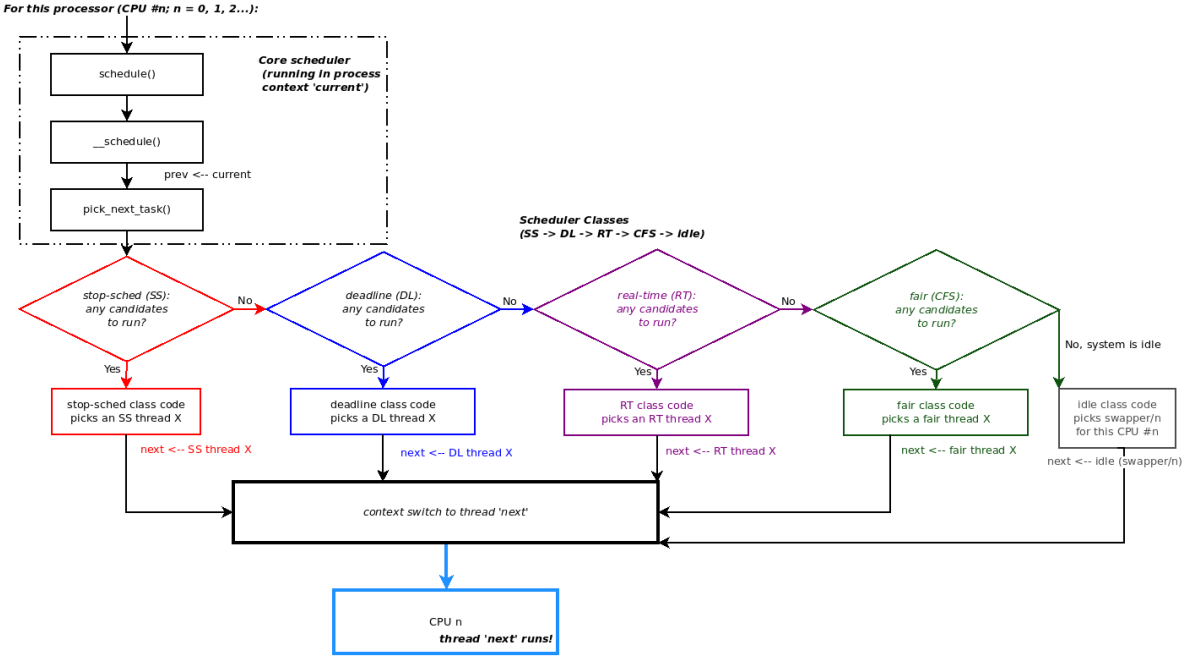
```
[  122.685801] oom_killer_try invoked oom-killer: gfp_mask=0x100cca(GFP_HIGHUSER_MOVABLE), order=0, oom_score_adj=0
[  122.685804] CPU: 0 PID: 2032 Comm: oom_killer_try Not tainted 5.4.0-llkd01 #2
[  122.685805] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[  122.685806] Call Trace:
[  122.685836]  dump_stack+0x66/0x90
[  122.685847]  dump_header+0x4a/0x27c
[  122.685853]  oom_kill_process.cold+0xb/0x10
[  122.685856]  out_of_memory+0x24d/0x4e0
[  122.685860]  __alloc_pages_slowpath+0xcf1/0xf60
[  122.685871]  __alloc_pages_nodemask+0x368/0x3b0
[  122.685875]  pagecache_get_page+0xc3/0x3a0
[  122.685878]  filemap_fault+0x70b/0xae0
[  122.685885]  ? ext4_filemap_fault+0x25/0x3f
[  122.685889]  ext4_filemap_fault+0x2d/0x3f
[  122.685896]  __do_fault+0x37/0x1a0
[  122.685899]  __handle_mm_fault+0x10b9/0x1ad0
[  122.685905]  handle_mm_fault+0x116/0x240
[  122.685908]  do_user_addr_fault+0x208/0x480
[  122.685912]  do_page_fault+0x31/0x190
[  122.685916]  page_fault+0x3e/0x50
```

Chapter 10: The CPU Scheduler - Part 1



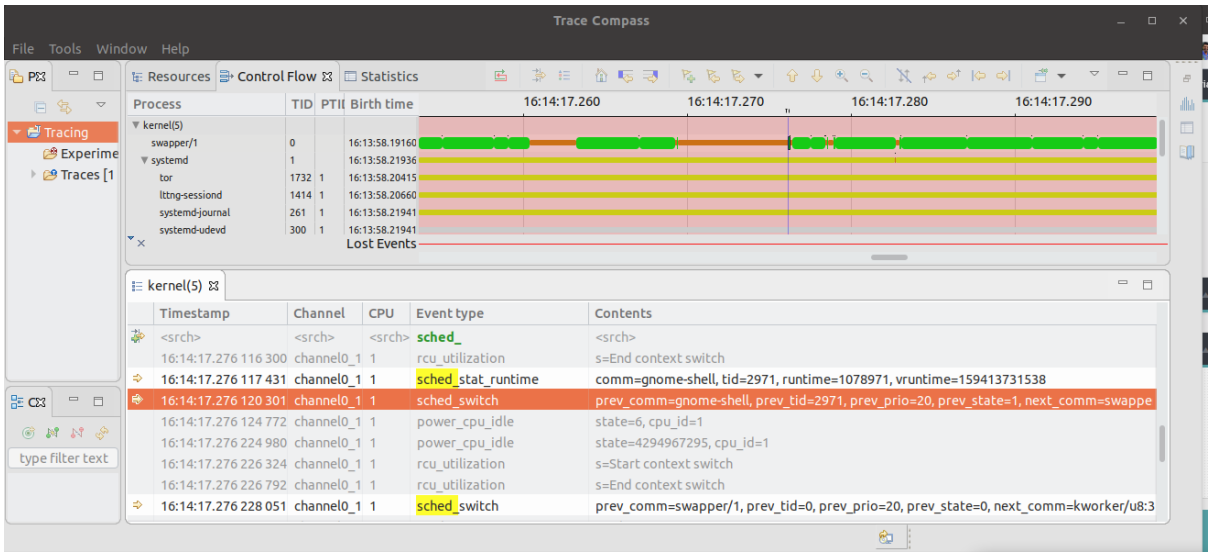



CPU 0
CPU 1
runqueue : stop-sched
runqueue : deadline
runqueue : RT (real-time)
runqueue : fair (CFS)
runqueue : idle
runqueue : stop-sched
runqueue : deadline
runqueue : RT (real-time)
runqueue : fair (CFS)
runqueue : idle


For this processor (CPU #n; n = 0, 1, 2...):
Core scheduler (running in process context 'current')
schedule()
__schedule()
prev <-- current
pick_next_task()
Scheduler Classes (SS -> DL -> RT -> CFS -> Idle)
stop-sched (SS): any candidates to run?
deadline (DL): any candidates to run?
real-time (RT): any candidates to run?
fair (CFS): any candidates to run?
No
Yes
No, system is idle
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!

```
ch10 $ ./query_task_sched.sh
  PID       TID            Name                    Sched Policy  Prio    *RT
    1         1                         systemd     SCHED_OTHER    0
    2         2                        kthreadd     SCHED_OTHER    0
    3         3                          rcu_gp     SCHED_OTHER    0
    4         4                      rcu_par_gp     SCHED_OTHER    0
    6         6                 kworker/0:0H-kblockd SCHED_OTHER    0
    9         9                      mm_percpu_wq     SCHED_OTHER    0
   10        10                       ksoftirqd/0     SCHED_OTHER    0
   11        11                         rcu_sched     SCHED_OTHER    0
   12        12                       migration/0      SCHED_FIFO   99      ***
   13        13                      idle_inject/0      SCHED_FIFO   50      *
   14        14                           cpuhp/0     SCHED_OTHER    0
   15        15                           cpuhp/1     SCHED_OTHER    0
   16        16                      idle_inject/1      SCHED_FIFO   50      *
   17        17                       migration/1      SCHED_FIFO   99      ***
   18        18                       ksoftirqd/1     SCHED_OTHER    0
   20        20                 kworker/1:0H-kblockd SCHED_OTHER    0
   21        21                           cpuhp/2     SCHED_OTHER    0
   22        22                      idle_inject/2      SCHED_FIFO   50      *
   23        23                       migration/2      SCHED_FIFO   99      ***
   24        24                       ksoftirqd/2     SCHED_OTHER    0
```

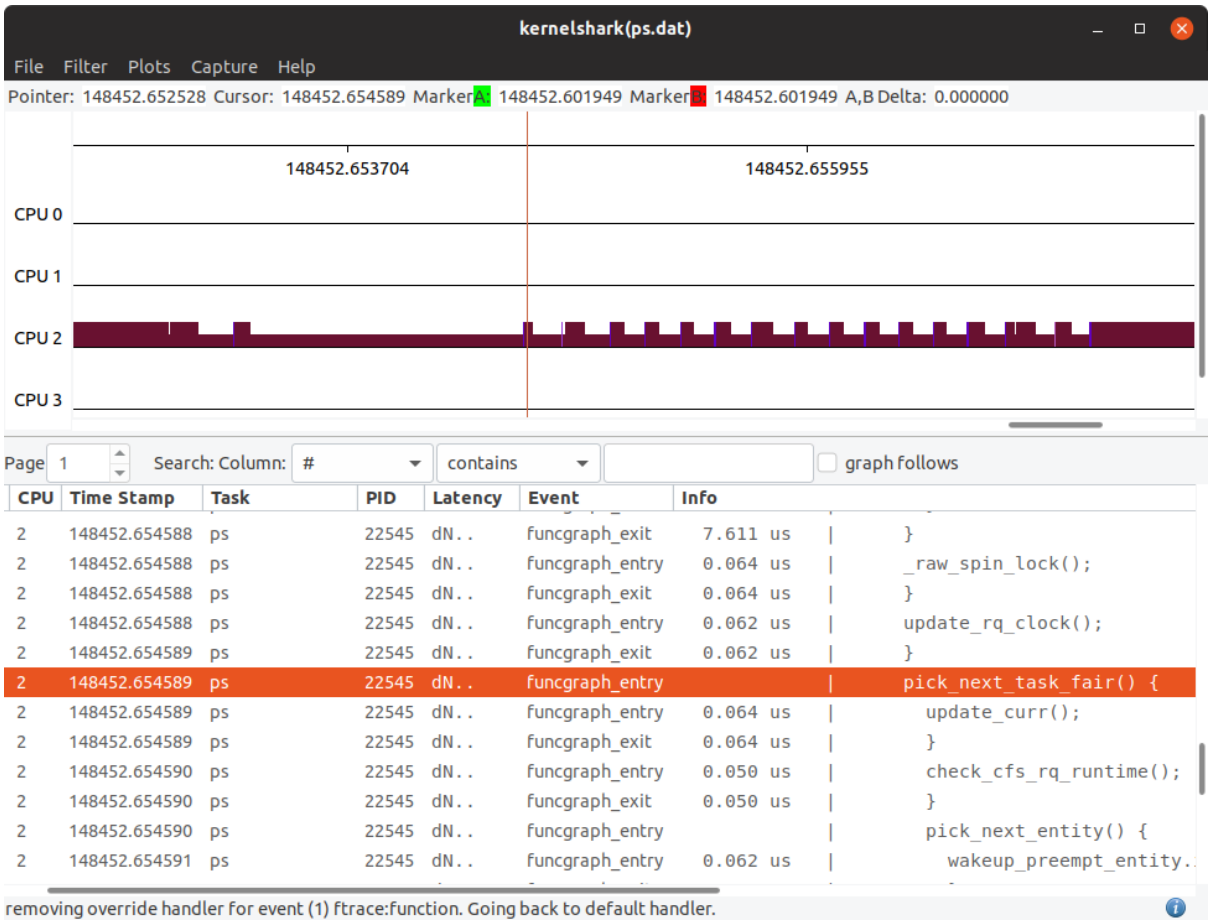
Chapter 11: The CPU Scheduler - Part 2




```
785299       ps-22922   2dN.. 149072.307624: funcgraph_entry:                    |  smp_reschedule_interrupt() {
785300       ps-22922   2dN.. 149072.307624: funcgraph_entry:        0.038 us    |    scheduler_ipi();
785301       ps-22922   2dN.. 149072.307625: funcgraph_exit:         0.531 us    |  }
785302       ps-22922   2dN.. 149072.307625: funcgraph_entry:                    |  exit_to_usermode_loop() {
785303       ps-22922   2.N.. 149072.307625: funcgraph_entry:                    |    schedule() {
785304       ps-22922   2dN.. 149072.307625: funcgraph_entry:                    |      rcu_note_context_switch() {
785305       ps-22922   2dN.. 149072.307626: funcgraph_entry:                    |        __event_probe__rcu_utilization() {
785306       ps-22922   2dN.. 149072.307626: funcgraph_entry:                    |          lttng_event_reserve() {
785307       ps-22922   2dN.. 149072.307626: funcgraph_entry:        0.077 us    |            ktime_get_mono_fast_ns();
```

```
#                     _-----=> irqs-off
#                    / _----=> need-resched
#                   | / _---=> hardirq/softirq
#                   || / _--=> preempt-depth
#                   ||| /
# CPU  TASK/PID     ||||  DURATION                  FUNCTION CALLS
# |     |    |      ||||   |   |                     |   |   |   |
 0)  kworker-2820  |  d..1  1.416 us    |                                        stack_access_ok();
```

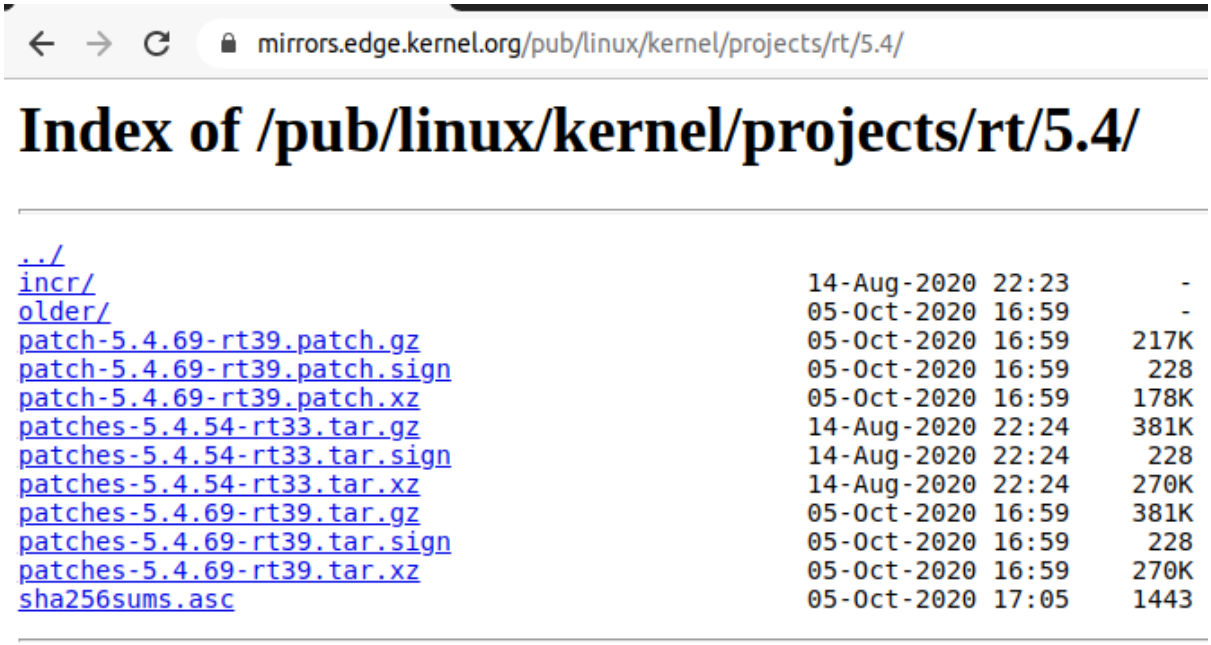
```
786463       ps-22922   2dN.. 149072.308038: funcgraph_entry:        0.054 us    |              update_rq_clock();
786464       ps-22922   2dN.. 149072.308038: funcgraph_entry:        0.221 us    |              pick_next_task_stop();
786465       ps-22922   2dN.. 149072.308039: funcgraph_entry:        0.053 us    |              pick_next_task_dl();
786466       ps-22922   2dN.. 149072.308040: funcgraph_entry:                    |              pick_next_task_rt() {
786467       ps-22922   2dN.. 149072.308040: funcgraph_entry:                    |                put_prev_task_fair() {
786468       ps-22922   2dN.. 149072.308040: funcgraph_entry:                    |                  put_prev_entity() {
786469       ps-22922   2dN.. 149072.308040: funcgraph_entry:                    |                    update_curr() {
786470       ps-22922   2dN.. 149072.308041: funcgraph_entry:        0.125 us    |                      update_min_vruntime();
```

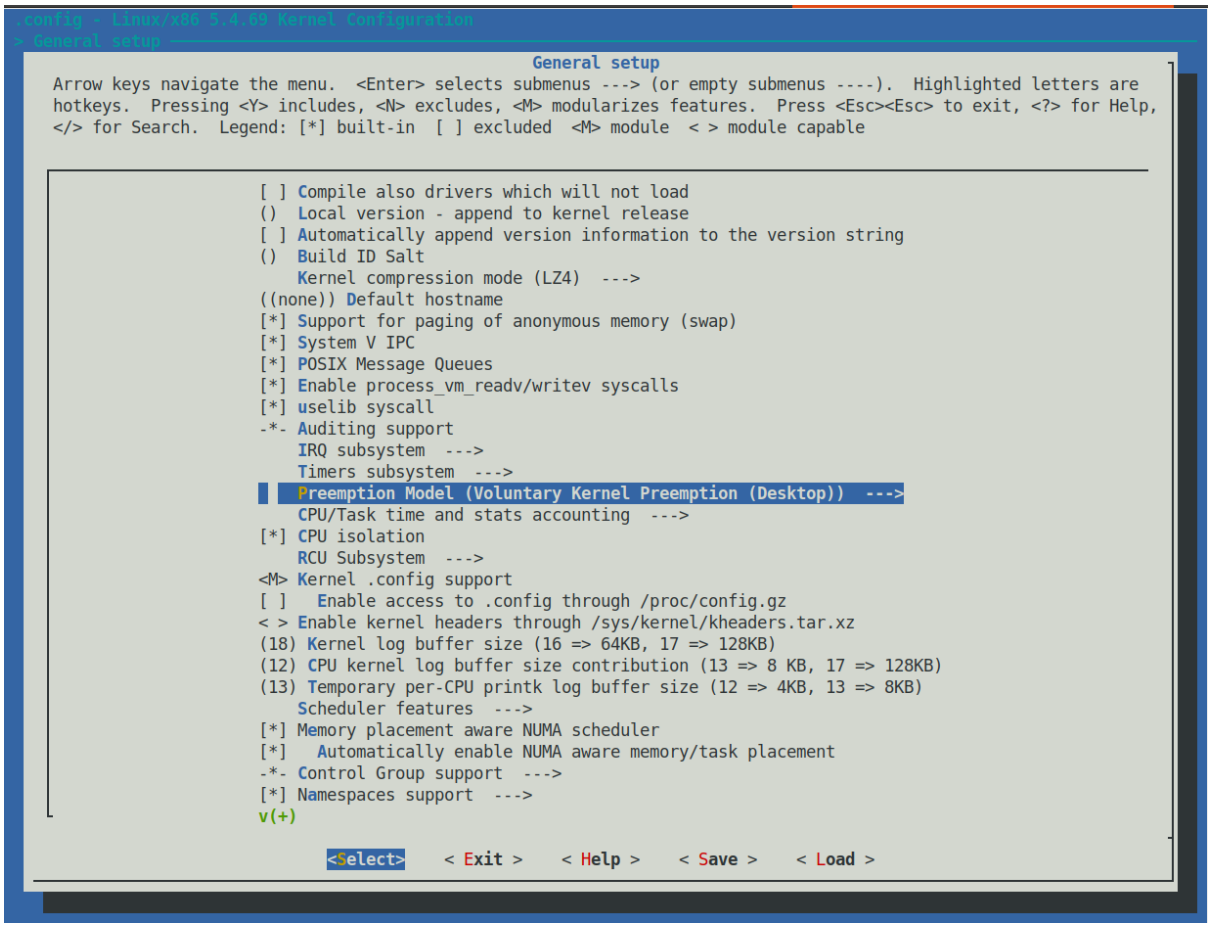


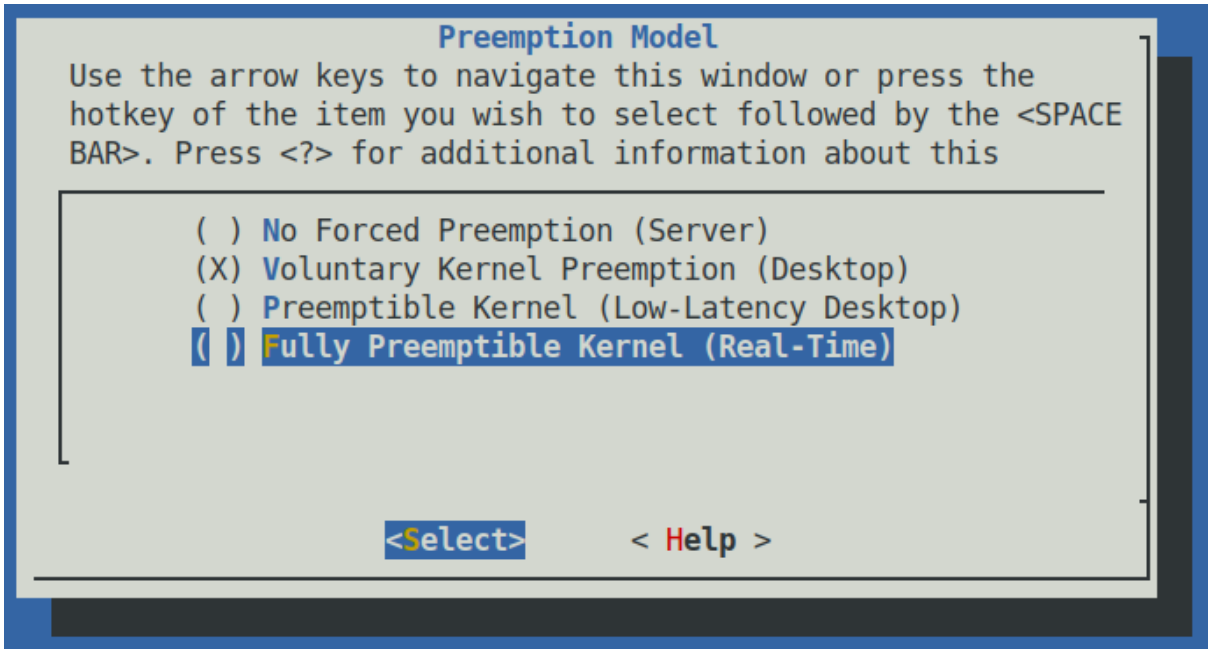

```
$ ./userspc_cpuaffinity
Detected 12 CPU cores [for this process ./userspc_cpuaffinity:335917]
CPU affinity mask for PID 335917:
 335917 pts/11   00:00:00 userspc_cpuaffi
       +--+--+--+--+--+--+--+--+--+--+--+--+
core#  |11|10| 9| 8| 7| 6| 5| 4| 3| 2| 1| 0|
       +--+--+--+--+--+--+--+--+--+--+--+--+
cpumask| 1| 1| 1| 1| 1| 1| 1| 1| 1| 1| 1| 1|
       +--+--+--+--+--+--+--+--+--+--+--+--+
$
```

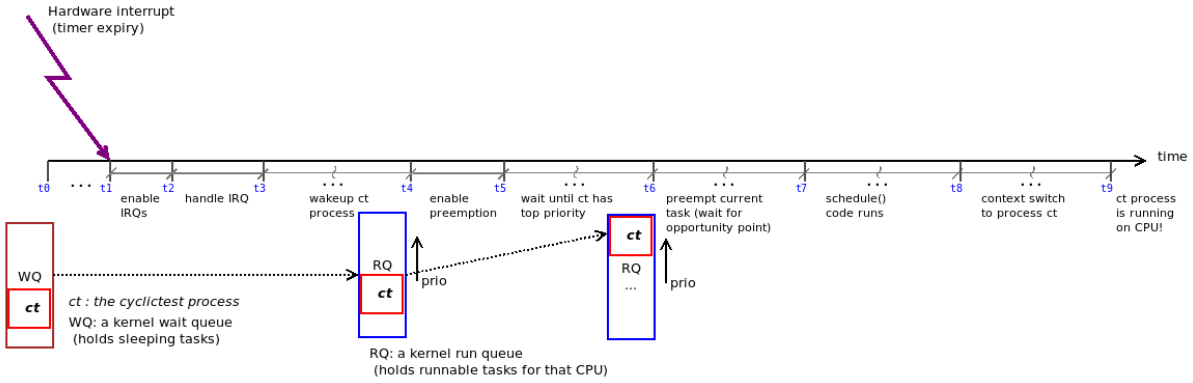
```
$ nproc
12
$ ps
    PID TTY          TIME CMD
 275621 pts/11   00:00:00 bash
 275896 pts/11   00:00:00 ps
$ ./userspc_cpuaffinity 275621 0xdae
Detected 12 CPU cores [for this process ./userspc_cpuaffinity:276018]
CPU affinity mask for PID 275621:
 275621 pts/11   00:00:00 bash
       +--+--+--+--+--+--+--+--+--+--+--+--+
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 275621 now...
CPU affinity mask for PID 275621:
 275621 pts/11   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|
       +--+--+--+--+--+--+--+--+--+--+--+--+
$
```

```
$ sudo ./cgv2_cpu_ctrl.sh 800000
[+] Checking for cgroup v2 kernel support
[+] Adding a 'cpu' controller to the cgroups v2 hierarchy
[+] Create a sub-group under it (here: /sys/fs/cgroup/test_group)

***
Now allowing 800000 out of a period of 1000000 by all processes (j1,j2) in this
sub-control group, i.e., 80.000% !
***

[+] Launch processes j1 and j2 (slinks to /home/llkd/Learn-Linux-Kernel-Development/ch11/cgroups_v2_cpu_eg/simp.sh
) now ...
[+] Insert processes j1 and j2 into our new CPU ctrl sub-group
Verifying their presence...
0::/test_group
Job j1 is in our new cgroup v2 test_group
0::/test_group
Job j2 is in our new cgroup v2 test_group

.............. sleep for 5 s ..............

[+] killing processes j1, j2 ...
./cgv2_cpu_ctrl.sh: line 185:  8805 Killed                  ./j1 1 > ${OUT1}
./cgv2_cpu_ctrl.sh: line 185:  8826 Killed                  ./j2 900 > ${OUT2}
cat 1stjob.txt
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68
cat 2ndjob.txt
900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 92
8 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956
957 958 959 960 961 962 963 964 965
[+] Removing our cpu sub-group controller
$
```



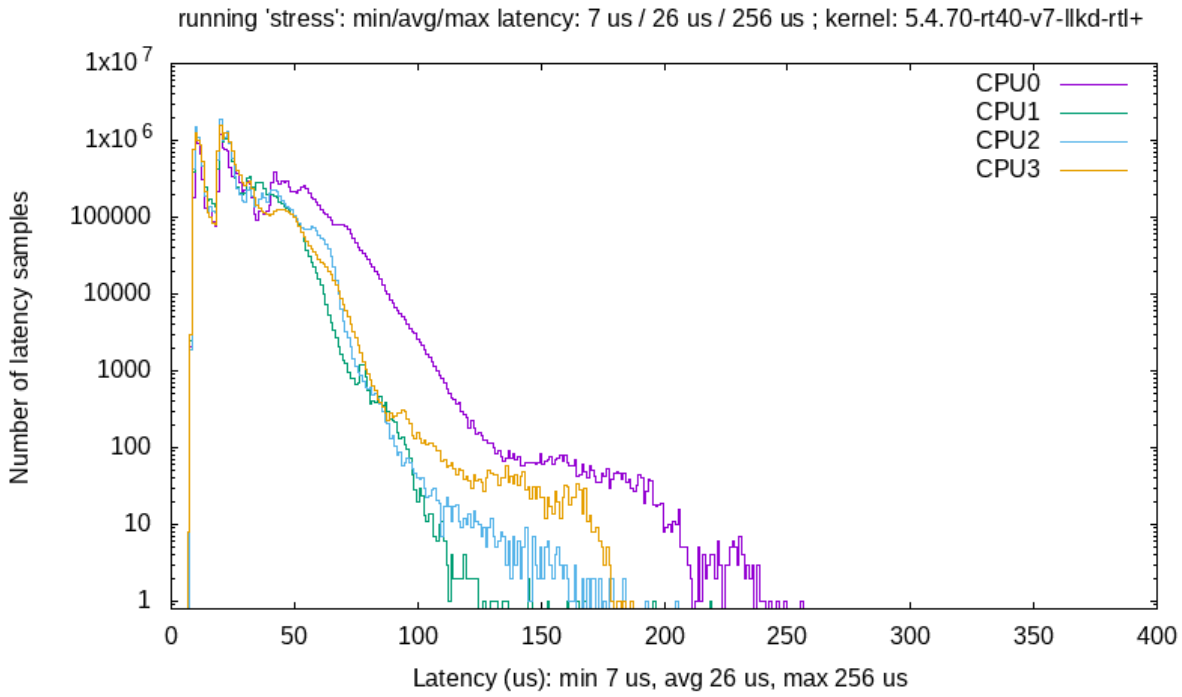

.config - Linux/x86 5.4.69 Kernel Configuration
> General setup
General setup
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable
[ ] Compile also drivers which will not load
() Local version - append to kernel release
[ ] Automatically append version information to the version string
() Build ID Salt
Kernel compression mode (LZ4) --->
((none)) Default hostname
[*] Support for paging of anonymous memory (swap)
[*] System V IPC
[*] POSIX Message Queues
[*] Enable process_vm_readv/writev syscalls
[*] uselib syscall
-*- Auditing support
IRQ subsystem --->
Timers subsystem --->
Preemption Model (Voluntary Kernel Preemption (Desktop)) --->
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)
Scheduler features --->
[*] Memory placement aware NUMA scheduler
[*] Automatically enable NUMA aware memory/task placement
-*- Control Group support --->
[*] Namespaces support --->
v(+)
<Select> < Exit > < Help > < Save > < Load >

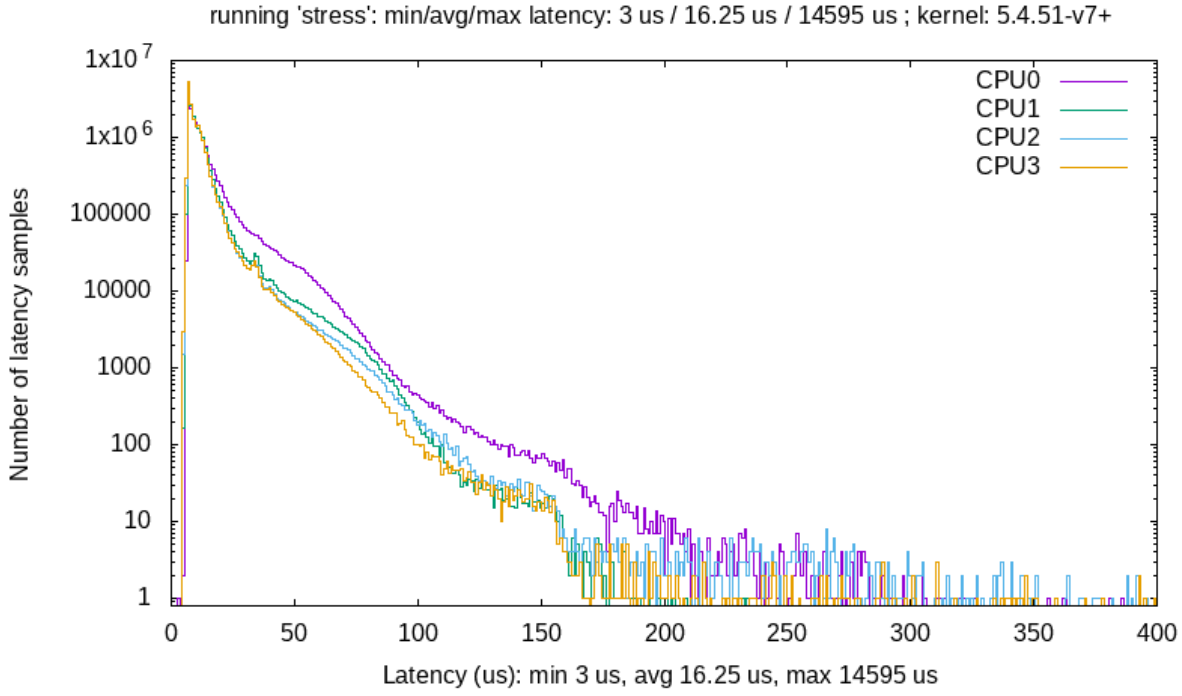

Preemption Model
Use the arrow keys to navigate this window or press the hotkey of the item you wish to select followed by the <SPACE BAR>. Press <?> for additional information about this
( ) No Forced Preemption (Server)
(X) Voluntary Kernel Preemption (Desktop)
( ) Preemptible Kernel (Low-Latency Desktop)
( ) Fully Preemptible Kernel (Real-Time)
<Select> < Help >

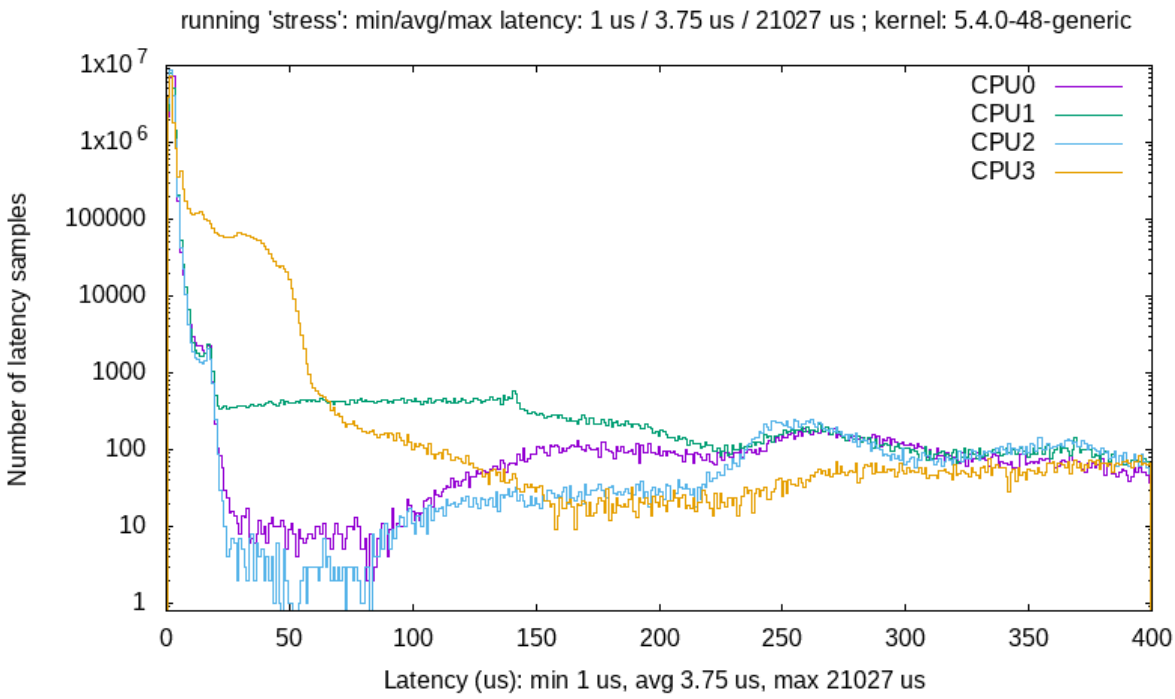



```
rpi latency_tests $ ./runtest
================================================================
Thu 15 Oct 11:58:19 IST 2020
stress --cpu 6 --io 2 --hdd 4 --hdd-bytes 1MB --vm 2 --vm-bytes 128M --timeout 1h
stress: info: [1059] dispatching hogs: 6 cpu, 2 io, 2 vm, 4 hdd
-------------------------
Test Title :: "running 'stress'"
-------------------------
Version info:
No LSB modules are available.
Distributor ID: Raspbian
Description:    Raspbian GNU/Linux 10 (buster)
Release:        10
Codename:       buster
Linux raspberrypi 5.4.70-rt40-v7-llkd-rtl+ #1 SMP PREEMPT_RT Thu Oct 15 07:58:13 IST 2020 armv7l GNU/Linux
Linux version 5.4.70-rt40-v7-llkd-rtl+ (kaiwan@kaiwan-7550) (gcc version 9.3.0 (Ubuntu 9.3.0-17ubuntu1~20.04)) #1 SMP PREEMPT_RT Thu Oct 15 07:
58:13 IST 2020

sudo /home/pi/rtl_llkd/rt-tests/cyclictest --duration=1h -m -Sp90 -i200 -h400 -q >output
stress: info: [1059] successful run completed in 3600s
Thu 15 Oct 12:58:19 IST 2020
Thu 15 Oct 12:58:19 IST 2020
rpi latency_tests $ min/avg/max latency: 7 us / 26 us / 256 us
```

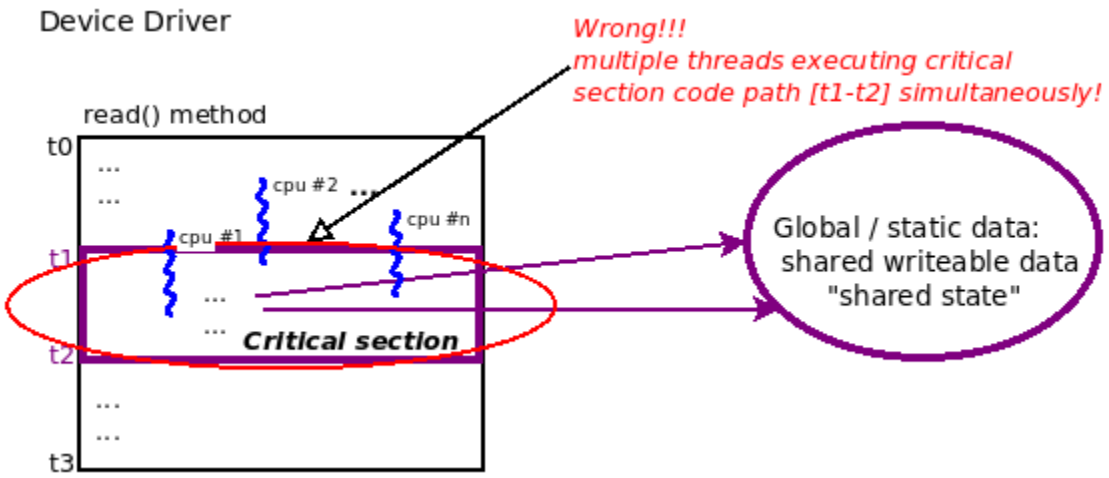
DUT (Device Under Test)	*System Latency (us)*		
	Min	*Avg*	*Max*
Raspberry Pi 3B+ ; 5.4.70-rt40 RTL kernel	7 us	26 us	256 us
Raspberry Pi 3B+ ; 5.4.51-v7+ standard kernel	3 us	16.3 us	14,595 us
x86_64 ; Ubuntu 20.04 5.4.0-48-generic standard kernel	1 us	3.8 us	21,027 us

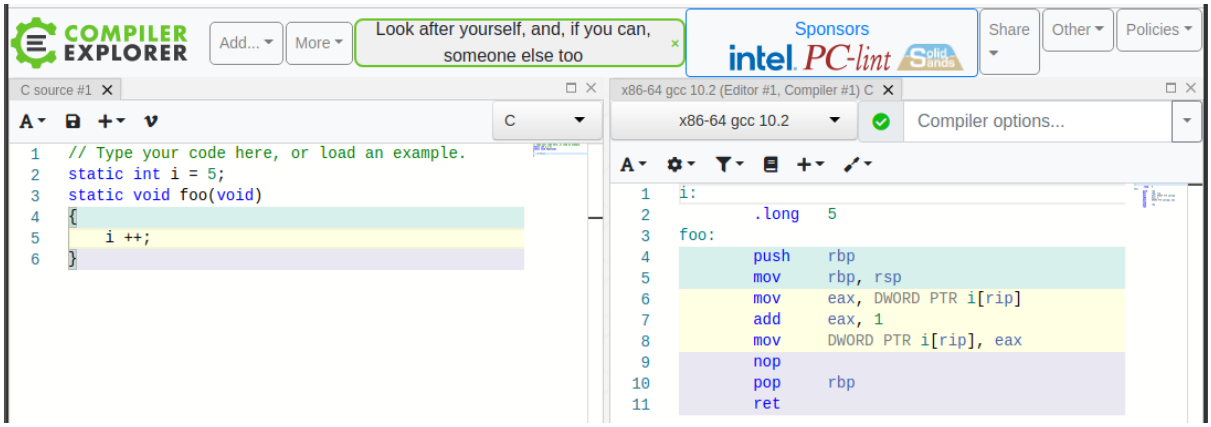

running 'stress': min/avg/max latency: 7 us / 26 us / 256 us ; kernel: 5.4.70-rt40-v7-llkd-rtl+
Number of latency samples
1x10 7
1x10 6
100000
10000
1000
100
10
1
0
50
100
150
200
250
300
350
400
Latency (us): min 7 us, avg 26 us, max 256 us
CPU0
CPU1
CPU2
CPU3

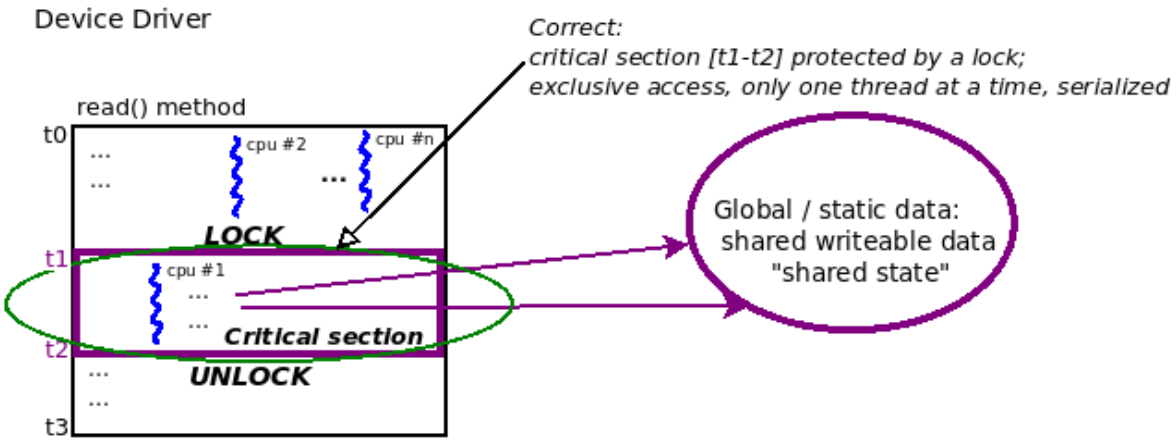

running 'stress': min/avg/max latency: 3 us / 16.25 us / 14595 us ; kernel: 5.4.51-v7+
Number of latency samples
1x10 7
1x10 6
100000
10000
1000
100
10
1
0
50
100
150
200
250
300
350
400
Latency (us): min 3 us, avg 16.25 us, max 14595 us
CPU0
CPU1
CPU2
CPU3

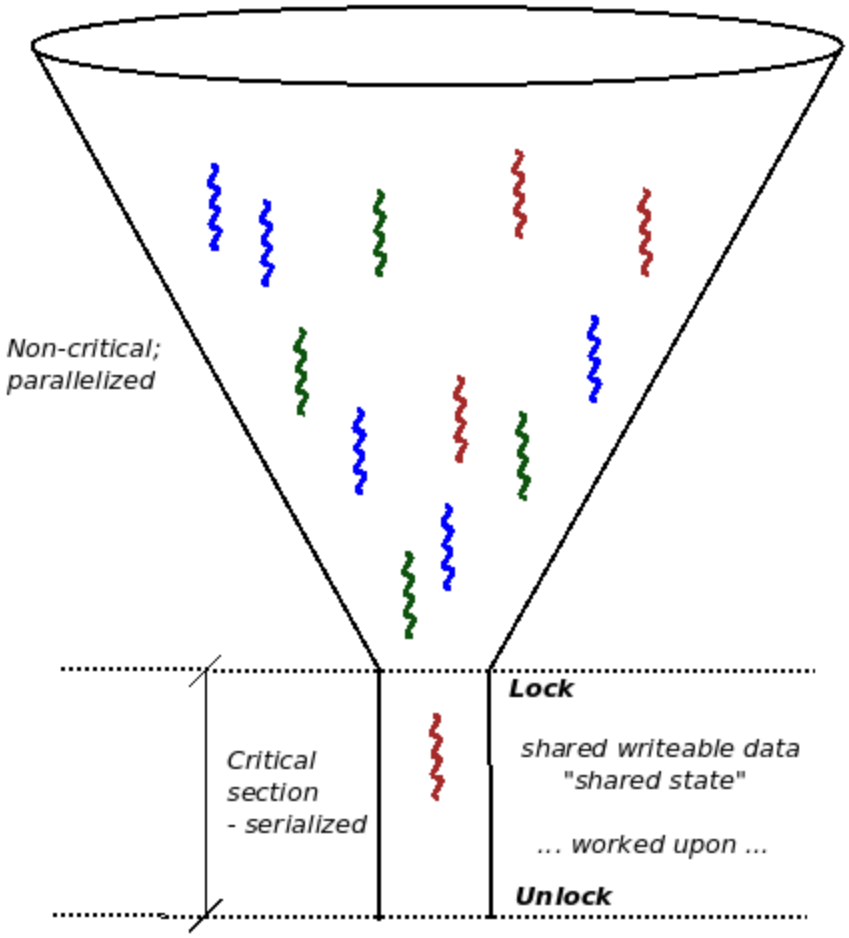

running 'stress': min/avg/max latency: 1 us / 3.75 us / 21027 us ; kernel: 5.4.0-48-generic
Number of latency samples
1x10 7
1x10 6
100000
10000
1000
100
10
1
0
50
100
150
200
250
300
350
400
Latency (us): min 1 us, avg 3.75 us, max 21027 us
CPU0
CPU1
CPU2
CPU3

Chapter 12: Kernel Synchronization - Part 1

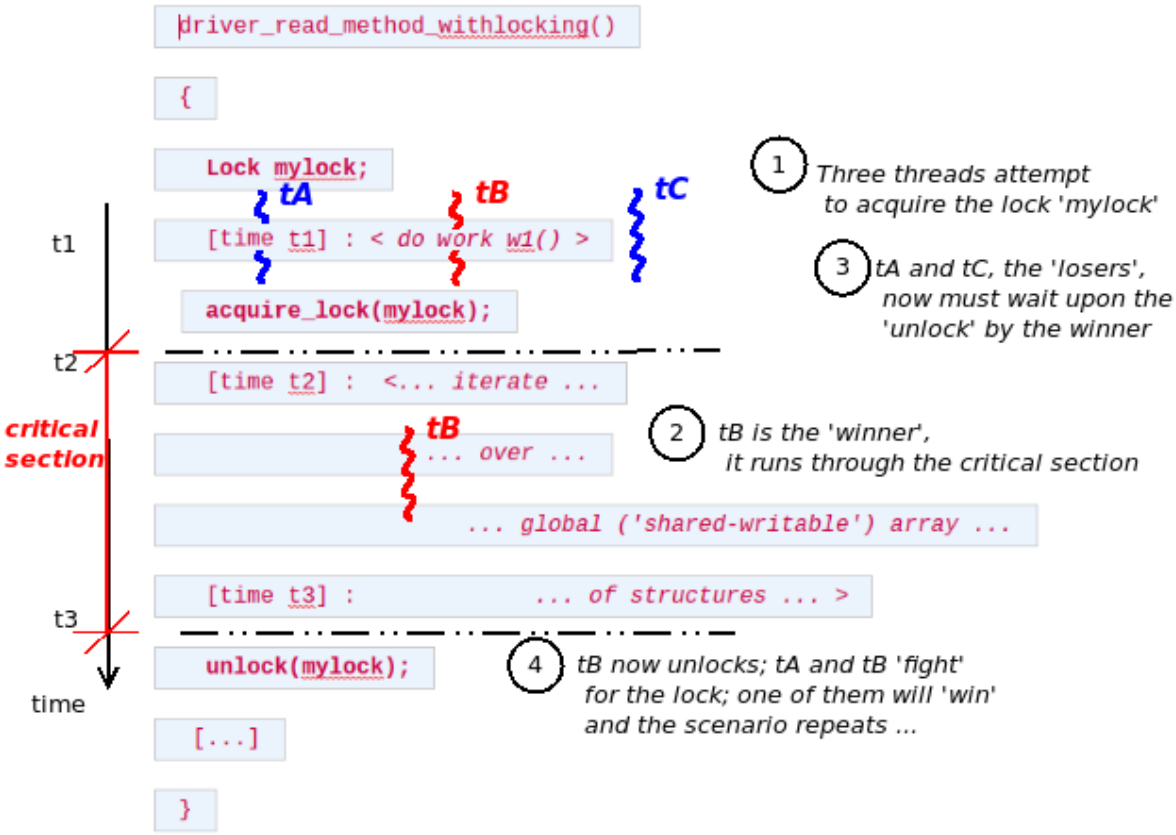




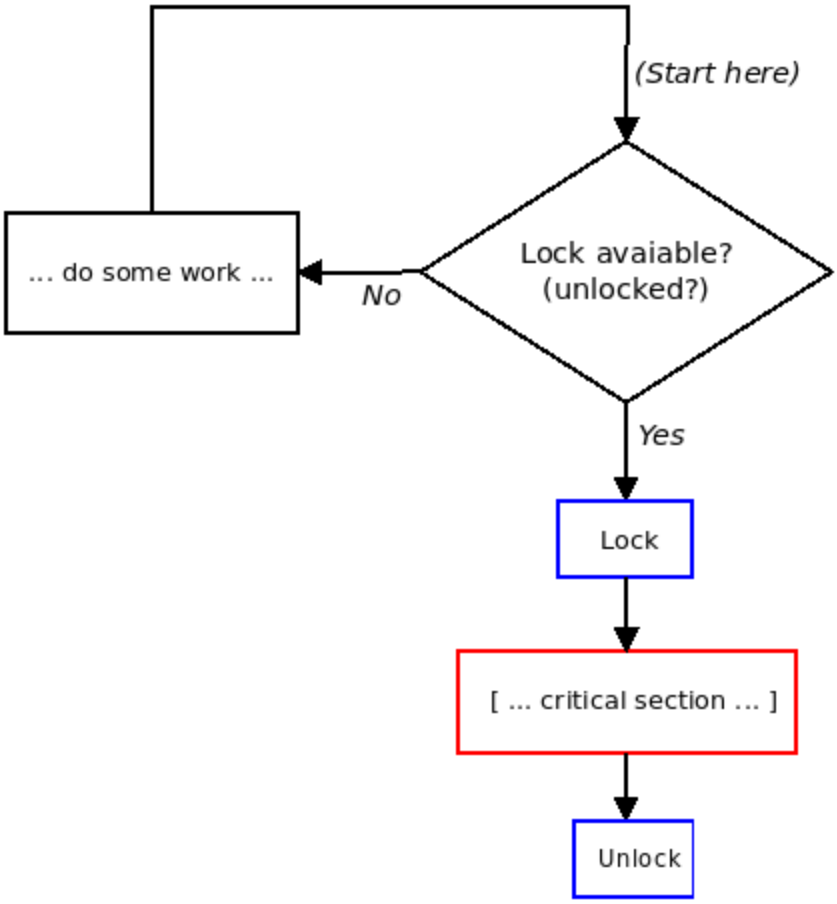



Non-critical;
parallelized
Lock
shared writeable data
"shared state"
... worked upon ...
Unlock
Critical
section
- serialized


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


driver_read_method_withlocking()
{
Lock mylock;
tA
tB
tC
t1
[time t1] : < do work w1() >
1
Three threads attempt
to acquire the lock 'mylock'
3
tA and tC, the 'losers',
now must wait upon the
'unlock' by the winner
acquire_lock(mylock);
t2
[time t2] : <... iterate ...
critical
section
tB
... over ...
2
tB is the 'winner',
it runs through the critical section
... global ('shared-writable') array ...
[time t3] : ... of structures ... >
t3
unlock(mylock);
4
tB now unlocks; tA and tB 'fight'
for the lock; one of them will 'win'
and the scenario repeats ...
time
[...]
}

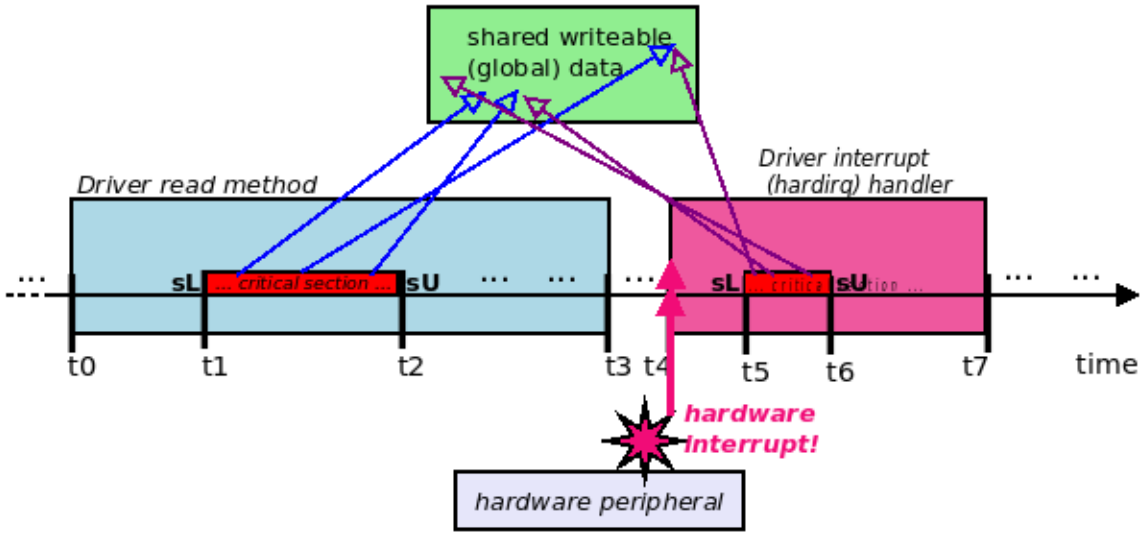
```
 static ssize_t read_miscdrv_rdwr(struct file *filp, char __user *ubuf,
-                size_t count, loff_t *off)
+               size_t count, loff_t *off)
 {
-   int ret = count, secret_len = strnlen(ctx->oursecret, MAXBYTES);
+   int ret = count, secret_len;
    struct device *dev = ctx->dev;

+   mutex_lock(&ctx->lock);
+   secret_len = strlen(ctx->oursecret);
+   mutex_unlock(&ctx->lock);
+
    PRINT_CTX();
    dev_info(dev, "%s wants to read (upto) %zd bytes\n", current->comm, count);

@@ -134,17 +140,20 @@
     * 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
+   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);
- out_notok:
+           secret_len, ctx->tx, ctx->rx);
+out_ctu:
+   mutex_unlock(&ctx->lock);
+out_notok:
    return ret;
```


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

```
[28853.172825] miscdrv_rdwr_spinlock:write_miscdrv_rdwr(): 004)  rdwr_test_secre :23578   |  ...0   /* write_mi
scdrv_rdwr() */
[28853.178231] misc llkd_miscdrv_rdwr_spinlock: rdwr_test_secre wants to write 24 bytes
[28853.181539] misc llkd_miscdrv_rdwr_spinlock:  24 bytes written, returning... (stats: tx=7, rx=24)
[28853.184243] BUG: scheduling while atomic: rdwr_test_secre/23578/0x00000002
[28853.187489] 1 lock held by rdwr_test_secre/23578:
[28853.189904]  #0: ffff8880285c2d60 (&(&ctx->spinlock)->rlock){+.+.}, at: write_miscdrv_rdwr.cold+0xde/0x247 [
miscdrv_rdwr_spinlock]
[28853.195078] Modules linked in: miscdrv_rdwr_spinlock(OE) vboxsf(OE) vboxvideo(OE) crct10dif_pclmul crc32_pcl
mul ghash_clmulni_intel vmwgfx snd_intel8x0 snd_ac97_codec ac97_bus snd_pcm aesni_intel glue_helper crypto_simd
 cryptd joydev snd_seq snd_timer drm_kms_helper snd_seq_device input_leds serio_raw snd syscopyarea sysfillrect
 sysimgblt fb_sys_fops ttm video mac_hid vboxguest(OE) soundcore drm sch_fq_codel parport_pc ppdev lp parport i
p_tables x_tables autofs4 hid_generic usbhid hid psmouse e1000 ahci libahci i2c_piix4 pata_acpi [last unloaded:
 miscdrv_rdwr_spinlock]
[28853.211613] CPU: 4 PID: 23578 Comm: rdwr_test_secre Tainted: G           OE     5.4.0-llkd-dbg #2
[28853.214596] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[28853.217244] Call Trace:
[28853.219461]  dump_stack+0xc2/0x11a
[28853.221692]  __schedule_bug.cold+0x2b/0x3c
[28853.223893]  __schedule+0xd4d/0x1090
[28853.226207]  ? firmware_map_remove+0xe9/0xe9
[28853.228428]  ? _raw_spin_unlock_irqrestore+0x51/0x60
[28853.230741]  ? schedule_timeout+0x2b4/0x8c0
[28853.232891]  ? lockdep_hardirqs_on+0x1a2/0x280
[28853.235050]  schedule+0x75/0x140
[28853.237118]  schedule_timeout+0x2b9/0x8c0
[28853.239207]  ? __dev_printk+0xd6/0xf3
[28853.241276]  ? usleep_range+0x100/0x100
[28853.243310]  ? _dev_info+0xcd/0xfb
[28853.245421]  ? __next_timer_interrupt+0xe0/0xe0
[28853.247475]  write_miscdrv_rdwr.cold+0x1ea/0x247 [miscdrv_rdwr_spinlock]
[28853.249726]  ? display_stats+0x80/0x80 [miscdrv_rdwr_spinlock]
[28853.251802]  ? apparmor_file_permission+0x1a/0x20
[28853.253814]  ? security_file_permission+0x65/0x190
[28853.255871]  __vfs_write+0x4f/0x90
[28853.257885]  vfs_write+0x14b/0x2d0
[28853.259744]  ksys_write+0xd9/0x180
[28853.261612]  ? __ia32_sys_read+0x50/0x50
[28853.263388]  ? mark_held_locks+0x29/0xb0
[28853.265119]  ? do_syscall_64+0x19/0x2c0
[28853.266842]  ? entry_SYSCALL_64_after_hwframe+0x49/0xbe
```

```
rdwr_tes-2438    4....  1060.741276: funcgraph_entry:                    |       vfs_write() {
rdwr_tes-2438    4....  1060.741276: funcgraph_entry:                    |         rw_verify_area() {
rdwr_tes-2438    4....  1060.741277: funcgraph_entry:                    |           security_file_permission() {
rdwr_tes-2438    4....  1060.741277: funcgraph_entry:                    |             apparmor_file_permission() {
rdwr_tes-2438    4....  1060.741277: funcgraph_entry:                    |               common_file_perm() {
rdwr_tes-2438    4....  1060.741277: funcgraph_entry:        0.244 us   |                 aa_file_perm();
rdwr_tes-2438    4....  1060.741277: funcgraph_exit:         0.492 us   |               }
rdwr_tes-2438    4....  1060.741277: funcgraph_exit:         0.715 us   |             }
rdwr_tes-2438    4....  1060.741278: funcgraph_exit:         1.010 us   |           }
rdwr_tes-2438    4....  1060.741278: funcgraph_exit:         1.273 us   |         }
rdwr_tes-2438    4....  1060.741278: funcgraph_entry:                    |         __vfs_write() {
rdwr_tes-2438    4....  1060.741278: funcgraph_entry:                    |           write_miscdrv_rdwr() {
rdwr_tes-2438    4....  1060.741278: funcgraph_entry:                    |             _dev_info() {
rdwr_tes-2438    4....  1060.741278: funcgraph_entry:                    |               __dev_printk() {
```

```
rdwr_tes-2438    4....  1060.746698: funcgraph_entry:                    |          schedule_timeout() {
rdwr_tes-2438    4....  1060.746698: funcgraph_entry:                    |            lock_timer_base() {
rdwr_tes-2438    4....  1060.746698: funcgraph_entry:        0.110 us   |              _raw_spin_lock_irqsave();
rdwr_tes-2438    4d...  1060.746698: funcgraph_exit:         0.318 us   |            }
rdwr_tes-2438    4d...  1060.746698: funcgraph_entry:        0.104 us   |            detach_if_pending();
rdwr_tes-2438    4d...  1060.746699: funcgraph_entry:        0.105 us   |            get_nohz_timer_target();
rdwr_tes-2438    4d...  1060.746699: funcgraph_entry:                    |            __internal_add_timer() {
rdwr_tes-2438    4d...  1060.746699: funcgraph_entry:        0.110 us   |              calc_wheel_index();
rdwr_tes-2438    4d...  1060.746699: funcgraph_entry:        0.161 us   |              enqueue_timer();
rdwr_tes-2438    4d...  1060.746699: funcgraph_exit:         0.588 us   |            }
rdwr_tes-2438    4d...  1060.746699: funcgraph_entry:        0.106 us   |            trigger_dyntick_cpu.isra.0();
rdwr_tes-2438    4d...  1060.746700: funcgraph_entry:        0.117 us   |              lock_text_start();
rdwr_tes-2438    4....  1060.746700: funcgraph_entry:                    |          schedule() {
rdwr_tes-2438    4d...  1060.746700: funcgraph_entry:                    |            rcu_note_context_switch() {
```



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

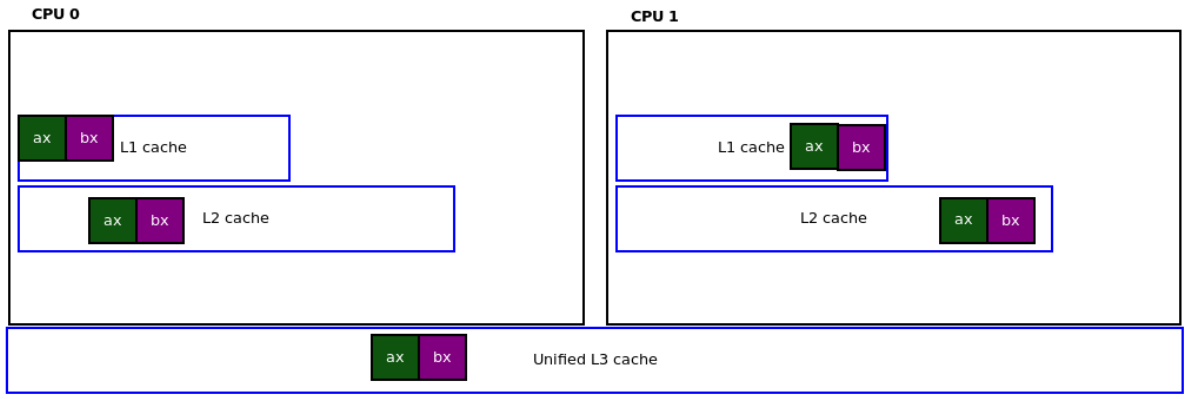
hardirq handler accessing shared writeable data

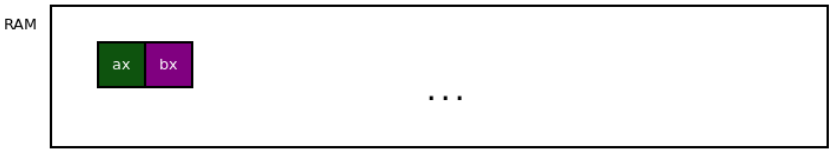
Chapter 13: Kernel Synchronization - Part 2

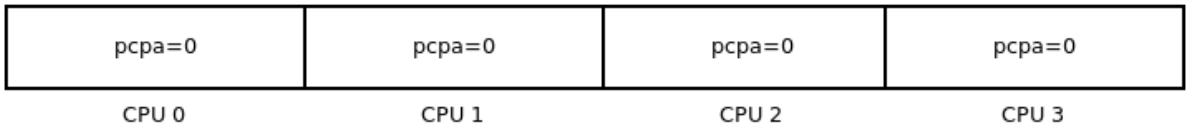
```
linux-5.4 $ grep -iHnA1 refcount kernel/user.c
kernel/user.c:100:      .__count        = REFCOUNT_INIT(1),
kernel/user.c-101-      .processes      = ATOMIC_INIT(1),
--
kernel/user.c:127:                      refcount_inc(&user->__count);
kernel/user.c-128-                      return user;
--
kernel/user.c:171:      if (refcount_dec_and_lock_irqsave(&up->__count, &uidhash_lock, &flags))
kernel/user.c-172-              free_user(up, flags);
--
kernel/user.c:190:              refcount_set(&new->__count, 1);
kernel/user.c-191-              ratelimit_state_init(&new->ratelimit, HZ, 100);
linux-5.4 $
```

```
$ dmesg
[ 7890.344169] miscdrv_rdwr_refcount:miscdrv_init_refcount(): LLKD misc driver (major # 10) registered, minor#
= 55, dev node is llkd_miscdrv_rdwr_refcount
[ 7890.345642] misc llkd_miscdrv_rdwr_refcount: A sample print via the dev_dbg(): driver initialized
[ 7904.871029] miscdrv_rdwr_refcount:open_miscdrv_rdwr(): 001)  rdwr_test_secre :8519   |  ...0   /* open_miscd
rv_rdwr() */
[ 7904.879384] ------------[ cut here ]------------
[ 7904.879735] refcount_t hit zero at open_miscdrv_rdwr+0x194/0x2b0 [miscdrv_rdwr_refcount] in rdwr_test_secre[
8519], uid/euid: 1001/1001
[ 7904.880685] WARNING: CPU: 1 PID: 8519 at kernel/panic.c:677 refcount_error_report+0xf1/0x103
[ 7904.881301] Modules linked in: miscdrv_rdwr_refcount(OE) vboxsf(OE) vboxvideo(OE) snd_intel8x0 vmwgfx snd_ac
97_codec ac97_bus snd_pcm crct10dif_pclmul crc32_pclmul ghash_clmulni_intel snd_seq aesni_intel glue_helper cry
pto_simd cryptd drm_kms_helper snd_timer snd_seq_device input_leds snd joydev syscopyarea serio_raw sysfillrect
 sysimgblt fb_sys_fops ttm soundcore vboxguest(OE) video mac_hid sch_fq_codel drm parport_pc ppdev lp parport i
p_tables x_tables autofs4 hid_generic usbhid hid psmouse e1000 ahci libahci i2c_piix4 pata_acpi [last unloaded:
 miscdrv_rdwr_refcount]
[ 7904.885282] CPU: 1 PID: 8519 Comm: rdwr_test_secre Tainted: G        W  OE     5.4.1-try1 #1
[ 7904.886040] Hardware name: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[ 7904.886668] RIP: 0010:refcount_error_report+0xf1/0x103
```

```
[15186.312399] 2_rmw_atomic_bitops: inserted
[15186.314690]  1:                    at init: mem :   0 = 0x00
[15186.315936]  2:            set_bit(7,&mem): mem : 128 = 0x80
[15186.317155] delta: 415 ns (= 0 us = 0 ms)
[15186.318746]  3: set msb suboptimal: 7,&mem: mem : 128 = 0x80
[15186.320096] delta: 110101 ns (= 110 us = 0 ms)
[15186.321285]  4:          clear_bit(7,&mem): mem :   0 = 0x00
[15186.323010]  5:         change_bit(7,&mem): mem : 128 = 0x80
[15186.324379]  6:   test_and_set_bit(0,&mem): mem : 129 = 0x81
[15186.325785]        ret = 0
[15186.327019]  7: test_and_clear_bit(0,&mem): mem : 128 = 0x80
[15186.328396]        ret (prev value of bit 0) = 1
[15186.329868]  8:test_and_change_bit(1,&mem): mem : 130 = 0x82
[15186.331487]        ret (prev value of bit 1) = 0
[15186.333013]  9: test_bit(7-0,&mem):
[15186.334436]   bit 7 (0x80) : set
[15186.335747]   bit 6 (0x40) : cleared
[15186.337013]   bit 5 (0x20) : cleared
[15186.338401]   bit 4 (0x10) : cleared
[15186.339648]   bit 3 (0x08) : cleared
[15186.340825]   bit 2 (0x04) : cleared
[15186.342129]   bit 1 (0x02) : set
[15186.343285]   bit 0 (0x01) : cleared
```

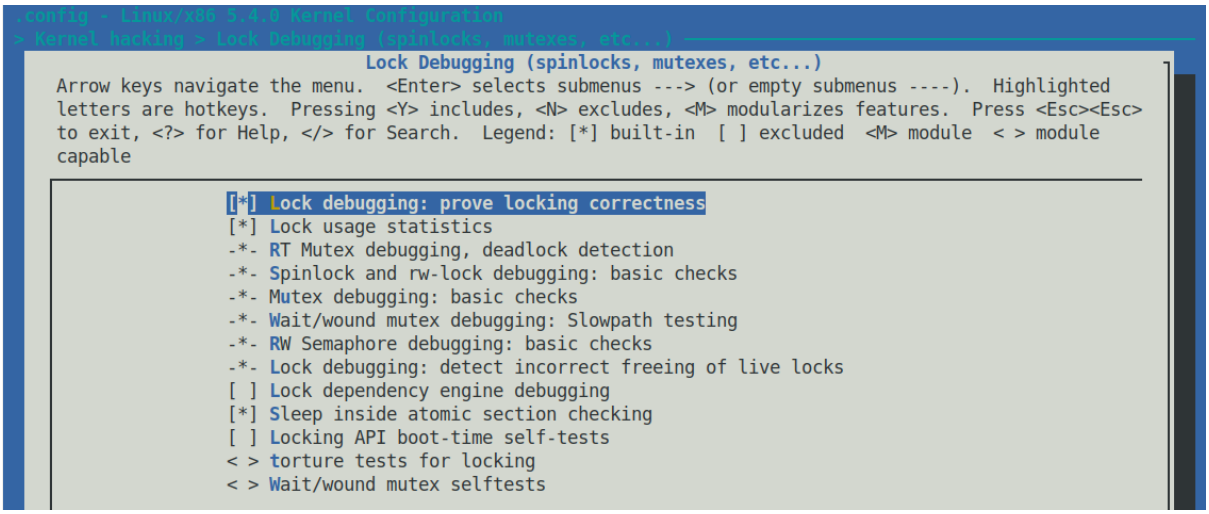







```
[ 2052.643407] percpu_var:init_percpu_var(): inserted
[ 2052.646162] percpu_var:thrd_work(): *** kthread PID 34971 on cpu 0 now ***
[ 2052.646648] percpu_var:thrd_work():   thrd_0/cpu0: pcpa = +1
[ 2052.647036] percpu_var:thrd_work():   thrd_0/cpu0: pcp ctx: tx =   100, rx =     0
[ 2052.647549] percpu_var:thrd_work():   thrd_0/cpu0: pcpa = +2
[ 2052.647942] percpu_var:thrd_work():   thrd_0/cpu0: pcp ctx: tx =   200, rx =     0
[ 2052.648506] percpu_var:thrd_work():   thrd_0/cpu0: pcpa = +3
[ 2052.648884] percpu_var:thrd_work():   thrd_0/cpu0: pcp ctx: tx =   300, rx =     0
[ 2052.649384] percpu_var:disp_vars(): 000) [thrd_0/0]:34971   |  .N.0   /* disp_vars() */
[ 2052.649979] percpu_var:disp_vars():  cpu  0: pcpa = +3, rx =     0, tx =   300
[ 2052.650486] percpu_var:disp_vars():  cpu  1: pcpa = +0, rx =     0, tx =     0
[ 2052.650999] percpu_var:thrd_work(): Our kernel thread #0 exiting now...
[ 2052.655130] percpu_var:thrd_work(): *** kthread PID 34972 on cpu 1 now ***
[ 2052.655750] percpu_var:thrd_work():   thrd_1/cpu1: pcpa = -1
[ 2052.656255] percpu_var:thrd_work():   thrd_1/cpu1: pcp ctx: tx =     0, rx =   200
[ 2052.656932] percpu_var:thrd_work():   thrd_1/cpu1: pcpa = -2
[ 2052.657440] percpu_var:thrd_work():   thrd_1/cpu1: pcp ctx: tx =     0, rx =   400
[ 2052.658275] percpu_var:thrd_work():   thrd_1/cpu1: pcpa = -3
[ 2052.658746] percpu_var:thrd_work():   thrd_1/cpu1: pcp ctx: tx =     0, rx =   600
[ 2052.659370] percpu_var:disp_vars(): 001) [thrd_1/1]:34972   |  .N.0   /* disp_vars() */
[ 2052.660051] percpu_var:disp_vars():  cpu  0: pcpa = +3, rx =     0, tx =   300
[ 2052.660684] percpu_var:disp_vars():  cpu  1: pcpa = -3, rx =   600, tx =     0
[ 2052.661280] percpu_var:thrd_work(): Our kernel thread #1 exiting now...
```

```
Functions calling this function: __alloc_percpu

  File              Function                Line
0 blk-stat.c        blk_stat_alloc_callback  118 cb->cpu_stat = __alloc_percpu(buckets * sizeof(struct blk_rq_stat),
1 blk-throttle.c    blk_throtl_init         2379 td->latency_buckets[READ] = __alloc_percpu(sizeof(struct latency_bucket) *
2 blk-throttle.c    blk_throtl_init         2385 td->latency_buckets[WRITE] = __alloc_percpu(sizeof(struct latency_bucket) *
3 devres.c          __devm_alloc_percpu     1087 pcpu = __alloc_percpu(size, align);
4 iova.c            init_iova_rcaches        871 rcache->cpu_rcaches = __alloc_percpu(sizeof(*cpu_rcache), cache_line_size());
5 irq-gic.c         gic_pm_init              771 gic->saved_ppi_enable = __alloc_percpu(DIV_ROUND_UP(32, 32) * 4,
6 irq-gic.c         gic_pm_init              776 gic->saved_ppi_active = __alloc_percpu(DIV_ROUND_UP(32, 32) * 4,
7 irq-gic.c         gic_pm_init              781 gic->saved_ppi_conf = __alloc_percpu(DIV_ROUND_UP(32, 16) * 4,
8 libcxgb_ppm.c     ppm_alloc_cpu_pool       369 pools = __alloc_percpu(alloc_sz, __alignof__(struct cxgbi_ppm_pool));
9 fc_exch.c         bool                    2503 mp->pool = __alloc_percpu(pool_size, __alignof__(struct fc_exch_pool));
a percpu.h          bool                     135 extern void __percpu *__alloc_percpu(size_t size, size_t align);
b percpu.h          alloc_percpu             143 (typeof(type) __percpu *)__alloc_percpu(sizeof(type), \
c kexec_core.c      crash_notes_memory_init 1105 crash_notes = __alloc_percpu(size, align);
d blktrace.c        do_blk_trace_setup       506 bt->msg_data = __alloc_percpu(BLK_TN_MAX_MSG, __alignof__(char ));
e blktrace.c        blk_trace_setup_queue   1609 bt->msg_data = __alloc_percpu(BLK_TN_MAX_MSG, __alignof__(char ));
f test_vmalloc.c    pcpu_alloc_test          318 pcpu[i] = __alloc_percpu(size, align);
g slab.c            alloc_kmem_cache_cpus   1729 cpu_cache = __alloc_percpu(size, sizeof(void *));
h slab.c            alloc_kmem_cache_cpus   3344 s->cpu_slab = __alloc_percpu(sizeof(struct kmem_cache_cpu),
i z3fold.c          z3fold_create_pool       781 pool->unbuddied = __alloc_percpu(sizeof(struct list_head)*NCHUNKS, 2);
j soft-interface.c  batadv_softif_init_late  762 bat_priv->bat_counters = __alloc_percpu(cnt_len, __alignof__(u64));
k route.c           ip_rt_init              3473 ip_rt_acct = __alloc_percpu(256 * sizeof(struct ip_rt_acct), __alignof__(struct
                                                 ip_rt_acct));
l x_tables.c        xt_percpu_counter_alloc 1842 state->mem = __alloc_percpu(XT_PCPU_BLOCK_SIZE,
m cls_u32.c         u32_change              1035 n->pf = __alloc_percpu(size, __alignof__(struct tc_u32_pcnt));
```




```
[ 1021.429110] thrd_showall_buggy: inserted
[ 1021.431264] --------------------------------------------------------------------------------
                   TGID   PID         current        stack-start      Thread Name   MT? # thrds
               --------------------------------------------------------------------------------

[ 1021.440804] ============================================
[ 1021.442866] WARNING: possible recursive locking detected
[ 1021.445129] 5.4.0-llkd-dbg #2 Tainted: G           OE
[ 1021.447157] --------------------------------------------
[ 1021.449384] insmod/2367 is trying to acquire lock:
[ 1021.451361] ffff88805de73f08 (&(&p->alloc_lock)->rlock){+.+.}, at: __get_task_comm+0x28/0x50
[ 1021.453676]
               but task is already holding lock:
[ 1021.457365] ffff88805de73f08 (&(&p->alloc_lock)->rlock){+.+.}, at: showthrds_buggy+0x13e/0x6d1 [thrd_showall_buggy]
[ 1021.461623]
               other info that might help us debug this:
[ 1021.465332]  Possible unsafe locking scenario:

[ 1021.468871]        CPU0
[ 1021.470563]        ----
[ 1021.472349]   lock(&(&p->alloc_lock)->rlock);
[ 1021.474591]   lock(&(&p->alloc_lock)->rlock);
[ 1021.476870]
                *** DEADLOCK ***

[ 1021.482086]  May be due to missing lock nesting notation

[ 1021.485550] 1 lock held by insmod/2367:
[ 1021.487884]  #0: ffff88805de73f08 (&(&p->alloc_lock)->rlock){+.+.}, at: showthrds_buggy+0x13e/0x6d1 [thrd_showall_buggy]
```

```
-static int showthrds_buggy(void)
+static int showthrds_fixed(void)
 {
    struct task_struct *g, *t;  /* 'g' : process ptr; 't': thread ptr */
    int nr_thrds = 1, total = 0;
@@ -60,7 +58,7 @@
    read_lock(&tasklist_lock);
 #endif
    do_each_thread(g, t) {     /* 'g' : process ptr; 't': thread ptr */
-       task_lock(t);
+       task_lock(t);  /*** task lock taken here! ***/

        snprintf(buf, BUFMAX-1, "%6d %6d ", g->tgid, t->pid);

@@ -70,12 +68,21 @@
        snprintf(tmp, TMPMAX-1, "  0x%016lx", (unsigned long)t->stack);
        strncat(buf, tmp, TMPMAX);

+   /* In the 'buggy' ver of this code, LOCKDEP did catch a deadlock here !!
+    * (at the point that get_task_comm() was invoked).
+    * the reason: get_task_comm() attempts to take the very same lock
+    * that we just took above: 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);
-/*--- LOCKDEP catches a deadlock here !! ---*/
+       task_lock(t);
```

```
$ sudo ./lock_stats_demo.sh
[+] Checking that locking statistics config is enabled   [OK]
[+] clearing lock stats ...
[+] enabling lock stats ...
cat/proc/self/cmdline[+] disabling lock stats ...
                          class name    con-bounces    contentions   waittime-min   waittime-max waittime-total   waittime-avg    acq-bo
unces   acquisitions   holdtime-min   holdtime-max holdtime-total   holdtime-avg
                 dup_mmap_sem.rw_sem-R:             0              0           0.00           0.00           0.00           0.00
    0              1         627.78         627.78         627.78         627.78
                       &mm->mmap_sem/1:             0              0           0.00           0.00           0.00           0.00
    0              1         624.38         624.38         624.38         624.38
         &(&mm->page_table_lock)->rlock:             0              0           0.00           0.00           0.00           0.00
    0             21           0.34           0.77           9.73           0.46
                       tasklist_lock-W:             0              0           0.00           0.00           0.00           0.00
    2              3           2.14          20.39          29.36           9.79
                       tasklist_lock-R:             0              0           0.00           0.00           0.00           0.00
    1              3           0.38           2.51           3.45           1.15
                &(&p->alloc_lock)->rlock:             0              0           0.00           0.00           0.00           0.00
    2             15           0.32           1.63           8.67           0.58
                    &mapping->i_mmap_rwsem:             0              0           0.00           0.00           0.00           0.00
    9            104           0.33           2.87          63.88           0.61
                      &mm->mmap_sem#2-W:             0              0           0.00           0.00           0.00           0.00
    0             32           0.35         626.64         986.59          30.83
                      &mm->mmap_sem#2-R:             0              0           0.00           0.00           0.00           0.00
    1            328           0.21          51.52        1803.33           5.50
     mmu_notifier_invalidate_range_start:             0              0           0.00           0.00           0.00           0.00
    0             58           0.22           0.79          14.16           0.24
                      &mm->context.lock:             0              0           0.00           0.00           0.00           0.00
    0              1           0.53           0.53           0.53           0.53
               &(&mm->arg_lock)->rlock:             0              0           0.00           0.00           0.00           0.00
    0              2           0.40           0.61           1.01           0.51
                      &ei->i_mmap_sem-R:             0              0           0.00           0.00           0.00           0.00
    3              5           1.35           2.13           8.43           1.69
$
```