Linux kernel debugging techniques

Omkar Savagaonkar
Software Engineer @ Cisco(CSG-DCBU)

How do we proceed ?

- Problem definition
- How this can be tackled
- What technique we can use which is offered by kernel
- Quick look at what goes in background
- Little bit of digression with some debug stories and experiences

How do I know if code path is executed

- Some mechanism to output logs on device file or normal file
- printk(KERN_ERR “Err log\n”);
- /proc/sys/kernel/printk

```
#define KERN_EMERG      "<0>"   /* system is unusable                  */
#define KERN_ALERT      "<1>"   /* action must be taken immediately    */
#define KERN_CRIT       "<2>"   /* critical conditions                 */
#define KERN_ERR        "<3>"   /* error conditions                    */
#define KERN_WARNING    "<4>"   /* warning conditions                  */
#define KERN_NOTICE     "<5>"   /* normal but significant condition    */
#define KERN_INFO       "<6>"   /* informational                       */
#define KERN_DEBUG      "<7>"   /* debug-level messages                */
```

Where to avoid printk

- Background (logging, klogd, syslogd(port 514), dmesg)
- Slow consoles
- locking
- Not for fast paths
- IRQs
- trace_printk …

Knowing code flow quickly with dump_stack()

```
[    0.362976] [<c048e614>] (twlreg_probe+0x1c0/0x24c) from [<c02d012c>] (platform_drv_probe+0x18/0x1c)
[    0.363006] [<c02d012c>] (platform_drv_probe+0x18/0x1c) from [<c02cec98>] (driver_probe_device+0x98/0x218)
[    0.363037] [<c02cec98>] (driver_probe_device+0x98/0x218) from [<c02cd47c>] (bus_for_each_drv+0x60/0x8c)
[    0.363037] [<c02cd47c>] (bus_for_each_drv+0x60/0x8c) from [<c02cefac>] (device_attach+0x98/0xbc)
[    0.363067] [<c02cefac>] (device_attach+0x98/0xbc) from [<c02cdfd8>] (bus_probe_device+0x88/0xb0)
[    0.363067] [<c02cdfd8>] (bus_probe_device+0x88/0xb0) from [<c02ccdd8>] (device_add+0x594/0x664)
[    0.363098] [<c02ccdd8>] (device_add+0x594/0x664) from [<c02d0730>] (platform_device_add+0xf4/0x20c)
[    0.363098] [<c02d0730>] (platform_device_add+0xf4/0x20c) from [<c02e4564>] (T.476+0xec/0x134)
[    0.363128] [<c02e4564>] (T.476+0xec/0x134) from [<c02e4618>] (add_regulator+0x6c/0x7c)
[    0.363159] [<c02e4618>] (add_regulator+0x6c/0x7c) from [<c048f200>] (twl_probe+0x848/0xa84)
[    0.363159] [<c048f200>] (twl_probe+0x848/0xa84) from [<c0393e24>] (i2c_device_probe+0xac/0xe8)
[    0.363189] [<c0393e24>] (i2c_device_probe+0xac/0xe8) from [<c02cec98>] (driver_probe_device+0x98/0x218)
[    0.363189] [<c02cec98>] (driver_probe_device+0x98/0x218) from [<c02cd47c>] (bus_for_each_drv+0x60/0x8c)
[    0.363220] [<c02cd47c>] (bus_for_each_drv+0x60/0x8c) from [<c02cefac>] (device_attach+0x98/0xbc)
[    0.363220] [<c02cefac>] (device_attach+0x98/0xbc) from [<c02cdfd8>] (bus_probe_device+0x88/0xb0)
[    0.363250] [<c02cdfd8>] (bus_probe_device+0x88/0xb0) from [<c02ccdd8>] (device_add+0x594/0x664)
[    0.363250] [<c02ccdd8>] (device_add+0x594/0x664) from [<c03947d4>] (i2c_new_device+0xf8/0x174)
[    0.363281] [<c03947d4>] (i2c_new_device+0xf8/0x174) from [<c0394bf0>] (i2c_register_adapter+0x168/0x218)
[    0.363311] [<c0394bf0>] (i2c_register_adapter+0x168/0x218) from [<c0394e00>] (i2c_add_numbered_adapter+0xc4/0xdc)
[    0.363311] [<c0394e00>] (i2c_add_numbered_adapter+0xc4/0xdc) from [<c049591c>] (omap_i2c_probe+0x350/0x430)
[    0.363342] [<c049591c>] (omap_i2c_probe+0x350/0x430) from [<c02d012c>] (platform_drv_probe+0x18/0x1c)
[    0.363342] [<c02d012c>] (platform_drv_probe+0x18/0x1c) from [<c02cec98>] (driver_probe_device+0x98/0x218)
[    0.363372] [<c02cec98>] (driver_probe_device+0x98/0x218) from [<c02ceeac>] (__driver_attach+0x94/0x98)
[    0.363372] [<c02ceeac>] (__driver_attach+0x94/0x98) from [<c02cd4fc>] (bus_for_each_dev+0x54/0x80)
[    0.363403] [<c02cd4fc>] (bus_for_each_dev+0x54/0x80) from [<c02cdc2c>] (bus_add_driver+0xa4/0x2ac)
[    0.363433] [<c02cdc2c>] (bus_add_driver+0xa4/0x2ac) from [<c02cf498>] (driver_register+0x78/0x180)
[    0.363433] [<c02cf498>] (driver_register+0x78/0x180) from [<c0008648>] (do_one_initcall+0x34/0x17c)
[    0.363464] [<c0008648>] (do_one_initcall+0x34/0x17c) from [<c065b380>] (kernel_init+0xf4/0x1c4)
```

Some requirements and situations

- Something goes wrong and log should be enabled at that point
- Crash is reported but additional register dump is needed
- Monitoring certain data at runtime

- We need some mechanism capable of quick interaction between user space and kernel space
- Kernel offers procfs and sysfs

Procfs and sysfs

- Configuration and control
- Dumping of buffers and data
- Useful info about system status
- Almost replacing ioctls
- Most portable with proper udev rules

Control certain actions

```
/* App.c */

int monitor_fn()
{
        if (something_goes_wrong)
                write_to_proc_file(enable_status_regs);

}
```

```
/* driver.c */

proc_write(value)
{
        enable_status_regs = value;

}

print_log()
{
        if (enable_status_reg)
                output_log();
}
```

Dumping of buffers and data

- How regular is certain interrupt ?
- Health of a device
- Some device malfunctions
  - Dump Registers
  - Power state
  - Defaults/settings

System status

- Cpuinfo
- Meminfo
- Devices
- Modules
- Interrupts
- Processes
- Slabinfo
- Maps

/proc/meminfo

```
MemTotal:        3941416 kB
MemFree:         2371672 kB
Buffers:           92972 kB
Cached:           795560 kB
SwapCached:            0 kB
Active:           717144 kB
Inactive:         721084 kB
Active(anon):     550868 kB
Inactive(anon):   101116 kB
Active(file):     166276 kB
Inactive(file):   619968 kB
Unevictable:           0 kB
Mlocked:               0 kB
SwapTotal:       5859324 kB
SwapFree:        5859324 kB
Dirty:               152 kB
Writeback:             0 kB
AnonPages:        549696 kB
Mapped:           198272 kB
Shmem:            102296 kB
Slab:              67516 kB
SReclaimable:      41256 kB
SUnreclaim:        26260 kB
KernelStack:        3432 kB
PageTables:        29028 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:     7830032 kB
Committed_AS:    2382152 kB
VmallocTotal:   34359738367 kB
```

/proc/slabinfo

```
kmalloc-4194304      0      0 4194304    1 1024 : tunables    1    1    0 : slabdata      0      0      0
kmalloc-2097152      0      0 2097152    1  512 : tunables    1    1    0 : slabdata      0      0      0
kmalloc-1048576      0      0 1048576    1  256 : tunables    1    1    0 : slabdata      0      0      0
kmalloc-524288       0      0 524288    1  128 : tunables    1    1    0 : slabdata      0      0      0
kmalloc-262144       0      0 262144    1   64 : tunables    1    1    0 : slabdata      0      0      0
kmalloc-131072       1      1 131072    1   32 : tunables    8    4    0 : slabdata      1      1      0
kmalloc-65536        8      8  65536    1   16 : tunables    8    4    0 : slabdata      8      8      0
kmalloc-32768        2      2  32768    1    8 : tunables    8    4    0 : slabdata      2      2      0
kmalloc-16384       43     43  16384    1    4 : tunables    8    4    0 : slabdata     43     43      0
kmalloc-8192        34     34   8192    1    2 : tunables    8    4    0 : slabdata     34     34      0
kmalloc-4096       299    319   4096    1    1 : tunables   24   12    8 : slabdata    299    319     12
kmalloc-2048       567    582   2048    2    1 : tunables   24   12    8 : slabdata    291    291      0
kmalloc-1024      1421   1448   1024    4    1 : tunables   54   27    8 : slabdata    362    362      0
kmalloc-512        978    984    512    8    1 : tunables   54   27    8 : slabdata    123    123    108
kmalloc-256        986   1095    256   15    1 : tunables  120   60    8 : slabdata     73     73      0
kmalloc-192       2692   2700    192   20    1 : tunables  120   60    8 : slabdata    135    135      0
kmalloc-96        1074   1209    128   31    1 : tunables  120   60    8 : slabdata     39     39      0
kmalloc-64        6853   6960     64   60    1 : tunables  120   60    8 : slabdata    116    116      0
kmalloc-128       1214   1519    128   31    1 : tunables  120   60    8 : slabdata     49     49      0
kmalloc-32       12976  13108     32  113    1 : tunables  120   60    8 : slabdata    116    116      0
kmem_cache         207    240    256   15    1 : tunables  120   60    8 : slabdata     16     16      0
```

/proc/devices

```
Character devices:
  1 mem
  4 /dev/vc/0
  4 tty
  4 ttyS
  5 /dev/tty
  5 /dev/console
  5 /dev/ptmx
  6 lp
  7 vcs
 10 misc
 13 input
 21 sg
 29 fb
 81 video4linux
 99 ppdev
116 alsa
119 vmnet
128 ptm
136 pts
166 ttyACM
180 usb
189 usb_device
226 drm
248 media
249 firewire
250 ptp
251 pps
252 bsg
253 watchdog
254 rtc
```

/proc/interrupts

- System not responsive to certain event
- Is there interrupt from my device ?
- If yes something wrong with userspace
- If no, check driver, actual device
- Dump regs of device

```
root@ThinkPad-T420:~# cat /proc/interrupts
           CPU0       CPU1       CPU2       CPU3
  0:         24          0          0          0   IO-APIC-edge      timer
  1:      10512          0          0          0   IO-APIC-edge      i8042
  8:          1          0          0          0   IO-APIC-edge      rtc0
  9:       8080          0          0          0   IO-APIC-fasteoi   acpi
 12:     764646          0          0          0   IO-APIC-edge      i8042
 16:        264          0         27          0   IO-APIC-fasteoi   ehci_hcd:usb1, mmc0
 19:         16          0          0          0   IO-APIC-fasteoi   firewire_ohci
 23:         80          0          0          0   IO-APIC-fasteoi   ehci_hcd:usb2
 42:      38381          0       5357          0   PCI-MSI-edge      ahci
 43:         24          0          0          0   PCI-MSI-edge      mei_me
 44:       1837          0     579051          0   PCI-MSI-edge      iwlwifi
 45:     373318          0          0          0   PCI-MSI-edge      i915
 46:        771          0          0          0   PCI-MSI-edge      snd_hda_intel
 47:         18          0          0          0   PCI-MSI-edge      nouveau
NMI:         25         18         25         18   Non-maskable interrupts
```

It's worth having a look at /proc

```
1     17    23    2831  32    36    3721  3796  3897  46    7986  9065      cpuinfo     key-users    softirqs
10    18    2449  2855  3270  3666  3722  38    39    48    8     9066      crypto      kmsg         stat
106   1883  2488  2863  3271  3669  3726  3800  3902  5     8003  93        devices     kpagecount   swaps
11    1891  25    2893  3272  3670  3727  3803  391   5065  8133  9304      diskstats   kpageflags   sys
1185  1925  2576  2894  3273  3680  3729  3806  3912  5201  8314  9340      dma         loadavg      sysrq-trigger
1186  1942  2598  2895  3274  3682  3730  3821  3927  5415  8316  94        driver      locks        sysvipc
12    1955  26    29    3275  3686  3736  3826  3929  5449  8326  95        execdomains meminfo      timer_list
124   2     2616  2900  3278  3687  3742  383   3942  5546  8334  96        fb          misc         timer_stats
125   20    2625  2902  328   3689  3744  3833  3949  5715  8390  97        filesystems modules      tty
13    2055  2655  2925  33    3690  3763  3836  3956  6716  8432  98        fs          mounts       uptime
15    2085  2656  3     34    3694  3765  3857  3961  6863  8443  acpi      interrupts  mtrr         version
1519  21    2681  30    35    37    3767  3858  397   6935  8575  asound    iomem       net          vmallocinfo
1547  2104  2697  3050  3529  3701  3776  386   4074  7     86    buddyinfo ioports     pagetypeinfo vmnet
1552  2192  27    3082  3531  3704  3786  3865  4076  7334  87    bus       irq         partitions   vmstat
1554  22    2741  3091  3541  3707  3787  3868  4088  7912  88    cgroups   kallsyms    sched_debug  zoneinfo
1561  2256  2774  31    3588  3719  3791  3884  44    7913  9     cmdline   kcore       self
16    2298  28    3144  3590  3720  3792  389   449   7946  90    consoles  keys        slabinfo
```

Sysfs

- Mechanism to represent kernel objects, their attributes and their relationships with each other

Internal	External
Kernel Objects	Directories
Object Attributes	Regular files
Object relationships	Symbolic links

Sysfs structure

```
root[21:10:23]:/sys# tree -L 1
.
├── block
├── bus
├── class
├── dev
├── devices
├── firmware
├── fs
├── hypervisor
├── kernel
├── module
└── power

11 directories, 0 files
```

```
root[21:11:51]:/sys/bus# tree -L 1
.
├── acpi
├── clockevents
├── clocksource
├── cpu
├── event_source
├── firewire
├── i2c
├── machinecheck
├── media
├── mei
├── memory
├── mmc
├── node
├── pci
├── pci_express
├── platform
├── pnp
├── scsi
├── sdio
├── serio
├── spi
├── usb
├── workqueue
├── xen
└── xen-backend
```

Framework

```
root[21:18:36]:/sys/class# ls -l
total 0
drwxr-xr-x 2 root root 0 Feb 24 20:03 ata_device
drwxr-xr-x 2 root root 0 Feb 24 20:03 ata_link
drwxr-xr-x 2 root root 0 Feb 24 20:03 ata_port
drwxr-xr-x 2 root root 0 Feb 24 20:03 backlight
drwxr-xr-x 2 root root 0 Feb 24 20:03 bdi
drwxr-xr-x 2 root root 0 Feb 24 20:03 block
drwxr-xr-x 2 root root 0 Feb 24 20:03 bluetooth
drwxr-xr-x 2 root root 0 Feb 24 20:03 bsg
drwxr-xr-x 2 root root 0 Feb 24 20:03 dma
drwxr-xr-x 2 root root 0 Feb 24 20:03 dmi
drwxr-xr-x 2 root root 0 Feb 24 20:03 drm
drwxr-xr-x 2 root root 0 Feb 24 20:03 firmware
drwxr-xr-x 2 root root 0 Feb 24 20:03 graphics
drwxr-xr-x 2 root root 0 Feb 24 20:03 hwmon
drwxr-xr-x 2 root root 0 Feb 24 20:03 i2c-adapter
drwxr-xr-x 2 root root 0 Feb 24 20:03 ieee80211
drwxr-xr-x 2 root root 0 Feb 24 20:03 input
drwxr-xr-x 2 root root 0 Feb 24 20:03 leds
drwxr-xr-x 2 root root 0 Feb 24 20:03 mem
drwxr-xr-x 2 root root 0 Feb 24 20:03 misc
drwxr-xr-x 2 root root 0 Feb 24 20:03 mmc_host
drwxr-xr-x 2 root root 0 Feb 24 20:03 net
drwxr-xr-x 2 root root 0 Feb 24 20:03 pci_bus
drwxr-xr-x 2 root root 0 Feb 24 20:03 powercap
drwxr-xr-x 2 root root 0 Feb 24 20:03 power_supply
drwxr-xr-x 2 root root 0 Feb 24 20:03 ppdev
drwxr-xr-x 2 root root 0 Feb 24 20:03 pps
drwxr-xr-x 2 root root 0 Feb 24 20:03 printer
drwxr-xr-x 2 root root 0 Feb 24 20:03 ptp
```

- Device type
- Using framework APIs
- Ready sysfs entry
- Easy to debug with standard tools
- Evtest, i2c utilies, lspci, lsusb
- udev

/dev/mem interface

- This can be very useful if no proc or sys entries created and issue is observed in field
- All physical memory access is available
- Mmap the section of interest
- Dump particular section to get info
- Write to particular section to change the regs

And we got into oops ...

```
[ 1962.516052] Internal error: Oops - undefined instruction: 0 [#1] PREEMPT SMP
ARM
[ 1962.524169] Modules linked in: cls_fw sch_htb ip6table_filter ip6_tables
xt_multiport iptable_filter ip_tables x_tables cryptoloop omapdrm_pvr(O)
adv7604(O) drishti_v4l2_bridge(O)
[ 1962.541931] CPU: 0    Tainted: G           O  (3.4.19-75 #1)
[ 1962.547973] PC is at 0xd4c77af0
[ 1962.551330] LR is at 0xd619d7ec
[ 1962.554687] pc : [<d4c77af0>]    lr : [<d619d7ec>]    psr: 600f0013
[ 1962.554687] sp : d4b5bf10  ip : adc98000  fp : ad99a000
[ 1962.566894] r10: adeb8000  r9 : e83579f4  r8 : d4b6c840
[ 1962.572479] r7 : e83579f8  r6 : ebab68c0  r5 : 00000000  r4 : e8357a14
[ 1962.579437] r3 : ebab68fc  r2 : 008d0000  r1 : 008d0000  r0 : c0f8afc0
[ 1962.586334] Flags: nZCv  IRQs on  FIQs on  Mode SVC_32  ISA ARM  Segment
user
[ 1962.593933] Control: 10c5387d  Table: a839c04a  DAC: 00000015
[ 1962.600006]
[ 1962.600006] PC: 0xd4c77a70:
[ 1962.604675] 7a70  ff000000 ff000000 ff000000 ff000000 ff000000 ff000000
ff000000 ff000000
[ 1962.613983] 7a90  ff000000 ff000000 ff000000 ff000000 ff000000 ff000000
ff000000 ff000000
[ 1962.623260] 7ab0  ff000000 ff000000 ff000000 ff000000 ff000000 ff000000

[ 1962.679107] LR: 0xd619d76c:
```

Kernel text 0xC0004000-C00303FFF

```
[ 1962.516052] Internal error: Oops - undefined instruction: 0 [#1] PREEMPT SMP
ARM
[ 1962.524169] Modules linked in: cls_fw sch_htb ip6table_filter ip6_tables
xt_multiport iptable_filter ip_tables x_tables cryptoloop omapdrm_pvr(O)
adv7604(O) drishti_v4l2_bridge(O)
[ 1962.541931] CPU: 0    Tainted: G           O  (3.4.19-75 #1)
[ 1962.547973] PC is at 0xd4c77af0
[ 1962.551330] LR is at 0xd619d7ec
[ 1962.554687] pc : [<d4c77af0>]    lr : [<d619d7ec>]    psr: 600f0013
[ 1962.554687] sp : d4b5bf10  ip : adc98000  fp : ad99a000
[ 1962.566894] r10: adeb8000  r9 : e83579f4  r8 : d4b6c840
[ 1962.572479] r7 : e83579f8  r6 : ebab68c0  r5 : 00000000  r4 : e8357a14
[ 1962.579437] r3 : ebab68fc  r2 : 008d0000  r1 : 008d0000  r0 : c0f8afc0
[ 1962.586334] Flags: nZCv  IRQs on  FIQs on  Mode SVC_32  ISA ARM  Segment
user
[ 1962.593933] Control: 10c5387d  Table: a839c04a  DAC: 00000015
[ 1962.600006]
[ 1962.600006] PC: 0xd4c77a70:
[ 1962.604675] 7a70  ff000000 ff000000 ff000000 ff000000 ff000000 ff000000
ff000000 ff000000
[ 1962.613983] 7a90  ff000000 ff000000 ff000000 ff000000 ff000000 ff000000
ff000000 ff000000
[ 1962.623260] 7ab0  ff000000 ff000000 ff000000 ff000000 ff000000 ff000000

[ 1962.679107] LR: 0xd619d76c:
```

Can be the case of stack corruption ...

- CC_STACKPROTECTOR kernel config option under kernel features
- -fstack-protector feature of GCC
- Puts canary value on stack just before return address
- Value is validated before return
- Preventing buffer overflow
- Useful for us to detect stack corruption

Part of text corruption ...

- Write protecting kernel text section
- Writes from same processor are caught
- Not possible to catch writes from other processor in AMP
- DMA writes is also a problem

Another one ...

```
[    4.756835] Unable to handle kernel NULL pointer dereference at virtual address 000002b4
[    4.765380] pgd = c0004000
[    4.768188] [000002b4] *pgd=00000000
[    4.772003] Internal error: Oops: 5 [#1] SMP
[    4.776458] last sysfs file:
[    4.779541] Modules linked in:
[    4.782745] CPU: 0    Not tainted  (2.6.39.1-mg00.1 #1)
[    4.788208] PC is at blkdev_get+0x234/0x2cc
[    4.792572] LR is at blkdev_get+0x234/0x2cc
[    4.796936] pc : [<c0155088>]    lr : [<c0155088>]    psr: 60000013
[    4.796936] sp : cec2de20  ip : cec2ddf8  fp : 00000000
[    4.808898] r10: 00000000  r9 : ce802710  r8 : ffffffe2
[    4.814361] r7 : 00000083  r6 : c068a1ac  r5 : ce802700  r4 : ce802040
[    4.821166] r3 : 00000000  r2 : 00000000  r1 : 00000448  r0 : c063d5c0
[    4.827972] Flags: nZCv  IRQs on  FIQs on  Mode SVC_32  ISA ARM  Segment kernel
[    4.835601] Control: 10c5387d  Table: 80004019  DAC: 00000017
[    4.841613] Process swapper (pid: 1, stack limit = 0xcec2c2f8)
[    4.847686] Stack: (0xcec2de20 to 0xcec2e000)
```

Internal error: Oops: 5 [#1] SMP

- What's Oops 5 ?
- 5 ==> 0101 (User mode, read, protection fault)
- [#1] Occurred once

Bits	Bit is 0	Bit is 1
Bit [0]	No page found	Protection fault
Bit [1]	Read	Write
Bit [2]	Kernel mode	User mode

Stacktrace of Oops

```
[    4.980316] [<c0155088>] (blkdev_get+0x234/0x2cc) from [<c01551f8>] (blkdev_get_by_path+0x28/0x64)
[    4.989685] [<c01551f8>] (blkdev_get_by_path+0x28/0x64) from [<c012a3d0>] (mount_bdev+0x30/0x1a4)
[    4.998931] [<c012a3d0>] (mount_bdev+0x30/0x1a4) from [<c0187b88>] (ext3_mount+0x10/0x18)
[    5.007476] [<c0187b88>] (ext3_mount+0x10/0x18) from [<c012ad14>] (mount_fs+0x6c/0x168)
[    5.015838] [<c012ad14>] (mount_fs+0x6c/0x168) from [<c01415e0>] (vfs_kern_mount+0x4c/0x8c)
[    5.024566] [<c01415e0>] (vfs_kern_mount+0x4c/0x8c) from [<c01418f4>] (do_kern_mount+0x34/0xcc)
[    5.033660] [<c01418f4>] (do_kern_mount+0x34/0xcc) from [<c0143034>] (do_mount+0x644/0x6bc)
[    5.042388] [<c0143034>] (do_mount+0x644/0x6bc) from [<c014337c>] (sys_mount+0x84/0xc4)
[    5.050750] [<c014337c>] (sys_mount+0x84/0xc4) from [<c0008d10>] (do_mount_root+0x20/0xa8)
[    5.059387] [<c0008d10>] (do_mount_root+0x20/0xa8) from [<c0008e7c>] (mount_block_root+0xe4/0x1fc)
[    5.068725] [<c0008e7c>] (mount_block_root+0xe4/0x1fc) from [<c000912c>] (mount_root+0xa0/0xc0)
[    5.077819] [<c000912c>] (mount_root+0xa0/0xc0) from [<c00092b0>] (prepare_namespace+0x164/0x1c4)
[    5.087066] [<c00092b0>] (prepare_namespace+0x164/0x1c4) from [<c0008bac>] (kernel_init+0x114/0x154)
[    5.096618] [<c0008bac>] (kernel_init+0x114/0x154) from [<c005cf8c>] (kernel_thread_exit+0x0/0x8)
```

Observe the registers ...

```
[    4.756835] Unable to handle kernel NULL pointer dereference at virtual address 000002b4
[    4.765380] pgd = c0004000
[    4.768188] [000002b4] *pgd=00000000
[    4.772003] Internal error: Oops: 5 [#1] SMP
[    4.776458] last sysfs file:
[    4.779541] Modules linked in:
[    4.782745] CPU: 0    Not tainted  (2.6.39.1-mg00.1 #1)
[    4.788208] PC is at blkdev_get+0x234/0x2cc
[    4.792572] LR is at blkdev_get+0x234/0x2cc
[    4.796936] pc : [<c0155088>]    lr : [<c0155088>]    psr: 60000013
[    4.796936] sp : cec2de20  ip : cec2ddf8  fp : 00000000
[    4.808898] r10: 00000000  r9 : ce802710  r8 : ffffffe2
[    4.814361] r7 : 00000083  r6 : c068a1ac  r5 : ce802700  r4 : ce802040
[    4.821166] r3 : 00000000  r2 : 00000000  r1 : 00000448  r0 : c063d5c0
[    4.827972] Flags: nZCv  IRQs on  FIQs on  Mode SVC_32  ISA ARM  Segment kernel
[    4.835601] Control: 10c5387d  Table: 80004019  DAC: 00000017
[    4.841613] Process swapper (pid: 1, stack limit = 0xcec2c2f8)
[    4.847686] Stack: (0xcec2de20 to 0xcec2e000)
```

virtual address 000002b4, r1 : 00000448

```
struct big_struct {
        struct small1;
        struct small2;
        ...
        struct other;

};

struct other_struct {
        struct small;
        struct something;
        ...
        struct member;
};

other_fn(struct other_struct *other)
{
        /* Do something */
        other->member = value;
}


some_caller_fn(struct big_struct *big)
{
        other_fn(big->other_struct);
}
```

- (gdb) print &((other_struct *)0)->member
- $1 = (int *) 0x2b4
- What is r1: 448 ?
- Wrong address other struct
- Why ? Ans. big_struct

Let's debug this one

```
[  120.580169] CPU: 1 PID: 3052 Comm: insmod Tainted: P           OX 3.13.11-ckt13 #2
[  120.580255] Hardware name: LENOVO 4236M22/4236M22, BIOS 83ET66WW (1.36 ) 10/31/2011
[  120.580341] task: ffff880089f46000 ti: ffff880136c5c000 task.ti: ffff880136c5c000
[  120.580423] RIP: 0010:[<ffffffffa0006012>]  [<ffffffffa0006012>] oops_mod_init+0x12/0x1000 [oops]
[  120.580531] RSP: 0018:ffff880136c5dd30  EFLAGS: 00010296
[  120.580591] RAX: 000000000000000d RBX: 0000000000000000 RCX: 0000000000000000
[  120.580684] RDX: ffff88013e24f020 RSI: ffff88013e24d408 RDI: 0000000000000246
[  120.580762] RBP: ffff880136c5dd30 R08: 0000000000000082 R09: 00000000000003f2
[  120.580839] R10: 0000000000000000 R11: ffff880136c5da5e R12: ffffffffa0006000
[  120.580920] R13: ffffffffa0856000 R14: ffffffffa0856050 R15: 0000000000000001
[  120.580999] FS:  00007ff69c9cb740(0000) GS:ffff88013e240000(0000) knlGS:0000000000000000
[  120.581089] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[  120.581145] CR2: 0000000000000004 CR3: 0000000135849000 CR4: 00000000000407e0
[  120.581177] Stack:
[  120.581188]  ffff880136c5dda8 ffffffff8100214a 0000000000001000 ffffffffa0856000
[  120.581229]  ffffffffa0856050 0000000000000001 ffff880136c5dd80 ffffffff810597b3
[  120.581268]  0000000000000000 ffffffffa0008000 0000000085e274bc ffff880136c5def8
[  120.581309] Call Trace:
[  120.581327]  [<ffffffff8100214a>] do_one_initcall+0xfa/0x1b0
[  120.581356]  [<ffffffff810597b3>] ? set_memory_nx+0x43/0x50
[  120.581394]  [<ffffffff810e21b8>] load_module+0x2058/0x2700
[  120.581421]  [<ffffffff810ddb20>] ? store_uevent+0x40/0x40
[  120.581450]  [<ffffffff810e29d6>] SyS_finit_module+0x86/0xb0
[  120.581479]  [<ffffffff81730d2d>] system_call_fastpath+0x1a/0x1f
[  120.581506] Code: <c7> 04 25 04 00 00 00 00 00 00 00 31 c0 5d c3 00 00 00 00 00 00 00
[  120.581588] RIP  [<ffffffffa0006012>] oops_mod_init+0x12/0x1000 [oops]
[  120.581622]  RSP <ffff880136c5dd30>
[  120.581639] CR2: 0000000000000004
[  120.595973] ---[ end trace 4d130ad01b906504 ]---
```

.text address

```
root@ThinkPad-T420:~/test_oops# ls -al /sys/module/oops/sections/
total 0
drwxr-xr-x 2 root root    0 Feb 25 00:05 .
drwxr-xr-x 5 root root    0 Feb 25 00:03 ..
-r--r--r-- 1 root root 4096 Feb 25 00:05 .exit.text
-r--r--r-- 1 root root 4096 Feb 25 00:05 .gnu.linkonce.this_module
-r--r--r-- 1 root root 4096 Feb 25 00:05 .init.text
-r--r--r-- 1 root root 4096 Feb 25 00:05 __mcount_loc
-r--r--r-- 1 root root 4096 Feb 25 00:05 .note.gnu.build-id
-r--r--r-- 1 root root 4096 Feb 25 00:05 .rodata.str1.1
-r--r--r-- 1 root root 4096 Feb 25 00:05 .strtab
-r--r--r-- 1 root root 4096 Feb 25 00:05 .symtab
-r--r--r-- 1 root root 4096 Feb 25 00:05 .text
root@ThinkPad-T420:~/test_oops# cat /sys/module/oops/sections/.text
0xffffffffa0854000
```

Debugging with gdb

```
Reading symbols from oops.ko...done.
(gdb) add-symbol-file oops.o 0xffffffffa0854000
add symbol table from file "oops.o" at
        .text_addr = 0xffffffffa0854000
(y or n) y
Reading symbols from oops.o...done.
(gdb) disassemble oops
oops.c         oops.mod.c     oops_mod_exit  oops_mod_init
(gdb) disassemble oops_mod_init
Dump of assembler code for function oops_mod_init:
   0x0000000000000046 <+0>:     push   %rbp
   0x0000000000000047 <+1>:     mov    $0x0,%rdi
   0x000000000000004e <+8>:     xor    %eax,%eax
   0x0000000000000050 <+10>:    mov    %rsp,%rbp
   0x0000000000000053 <+13>:    callq  0x58 <oops_mod_init+18>
   0x0000000000000058 <+18>:    movl   $0x0,0x4
   0x0000000000000063 <+29>:    xor    %eax,%eax
   0x0000000000000065 <+31>:    pop    %rbp
   0x0000000000000066 <+32>:    retq
```

```c
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/init.h>

struct something {
        int i;
        int b;
};

void create_oops(struct something *) ;

static int __init oops_mod_init(void) {
        struct something *s = NULL;
        printk(KERN_WARNING "creating oops\n");
        create_oops(s);
        return (0);
}
static void __exit oops_mod_exit(void) {
        printk(KERN_NOTICE "No more oops\n");
}

void create_oops(struct something *s) {
        s = NULL;
        s->b = 0;
}

module_init(oops_mod_init);
module_exit(oops_mod_exit);
```

Some scenarios that are baffling

- There is sudden reboot after 17 hrs ...
- There is hard hang with no log on console
- There is crash with just few bytes of oops log

- We need some way to see what went wrong last time (reset reason)
- We need a mechanism which is capable of logging events fast

Two possibilities

- Something went wrong in hardware
- Check some persistent registers

- Something went wrong in software
- RAM console can be a friend
- Persistent tracer can give more (Ftrace)

Some experience ...

- All of a sudden system is unresponsive with no log
- Works fine with power cycle

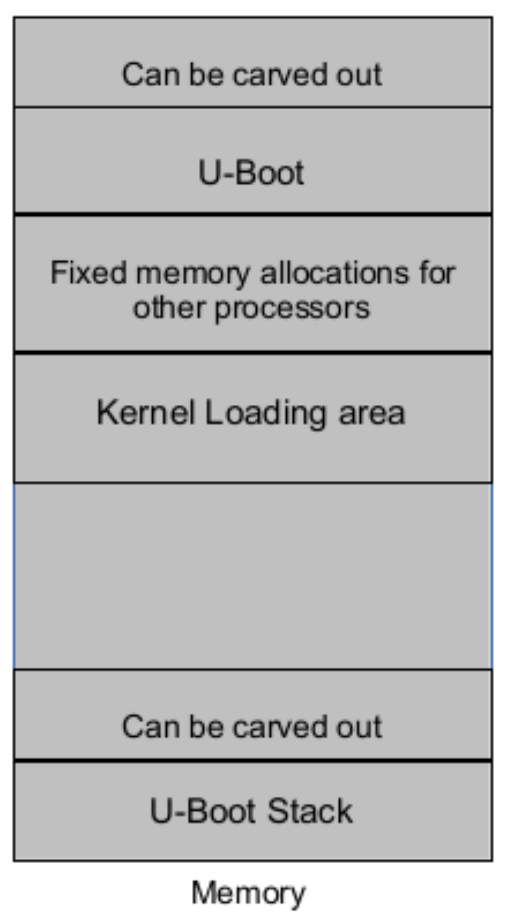
- Checked processor voltages and found to be dropped
- Power IC was still up !

Hardware is fine and it's software ...

- It's watchdog reset ..
- Memory logging
- Retrieving memory
- RAM console

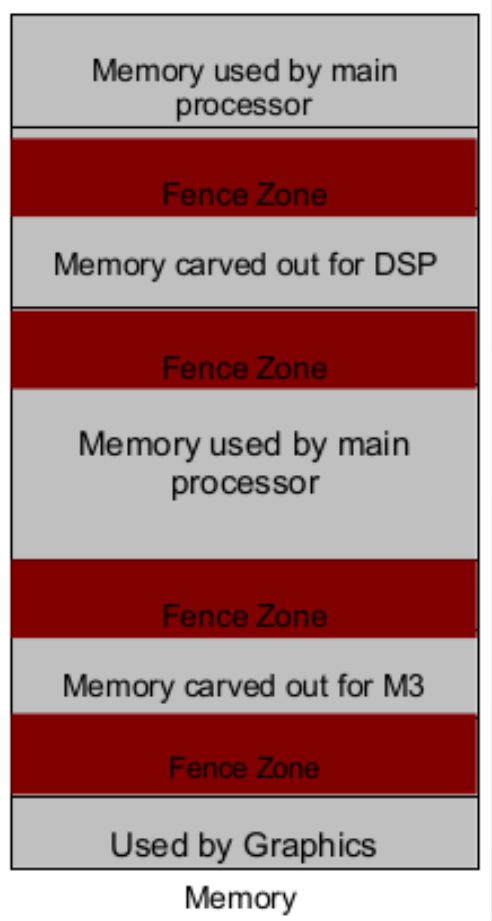
RAM console

- Persistent memory
- Bootloader an kernel loading area
- Register driver to console subsystem
- Write unique signature
- Log the messages
- Retrieving messages during next boot



Memory Fencing

- Carving out memory during early boot time
- Should not be the part of kernel or bootloader loading area
- Fence zones at multiple places based on memory allocation to different cores in SoC
- Check the pattern and signature during boot
- Boot reason is imp.
- Pattern tampered: Case of corruption



Kexec and kdump

- Kexec: used to boot different OS kernel
- Kdump: config for dumping core of kernel
- As good as debugging like core dump
- Loading regular system kernel
- Loading dump-capture kernel
- /proc/vmcore
- gdb vmlinux dump
- Another option is kgdb

Exploring hardware support if any

- Multiple asymmetric multiprocessing cores
- Different level of interfaces to access memory
- L3 interconnect
- Firewall supported per processor per memory region

Debugging performance issues

- Some process was ready but not scheduled
- Interrupt not being serviced at all
- Interrupt being serviced late
- Misleading scenarios due to interrupt and premption disabling

Playing with process and interrupt affinities

- Load balancing threads and processes on multicore system
- More the number of cores more is the chance to play and more headache
- Locking the application to memory
- There was interrupt but it never occurred on core0
- Setting interrupt affinities disabling irq balance for debug purpose

Ftrace

- Named after function tracer … but it's much more than that
- Major part developed by Steven Rostedt
- Under kernel config option of tracers
- Requires debugfs

```
--- Tracers
[*]   Kernel Function Tracer
[ ]     Kernel Function Graph Tracer
[*]   Interrupts-off Latency Tracer
[*]   Preemption-off Latency Tracer
[ ]   Scheduling Latency Tracer
      Branch Profiling (No branch profiling)  --->
[ ]   Trace max stack
[ ]   Support for tracing block IO actions
[*]   enable/disable function tracing dynamically
```

What ftrace can tell ?

- What functions of kernel are being executed during certain process contexts ?
- How processes are being scheduled ?
- sched_switch, sched_wakeup etc.
- Is there any irq latency ?
- Is there any latency due to preemption disabled ?
- Is there any scheduling latency ?
- What might have caused these latencies ?

/sys/kernel/debug/tracing

- Options available under tracing
- available_tracers
- available_events

```
root@ThinkPad-T420:/sys/kernel/debug/tracing# ls
available_events            free_buffer                 README              stack_max_size      tracing_cpumask
available_filter_functions  function_profile_enabled    saved_cmdlines      stack_trace         tracing_max_latency
available_tracers           instances                   set_event           stack_trace_filter  tracing_on
buffer_size_kb              kprobe_events               set_ftrace_filter   trace               tracing_thresh
buffer_total_size_kb        kprobe_profile              set_ftrace_notrace  trace_clock         uprobe_events
current_tracer              max_graph_depth             set_ftrace_pid      trace_marker        uprobe_profile
dyn_ftrace_total_info       options                     set_graph_function  trace_options
enabled_functions           per_cpu                     set_graph_notrace   trace_pipe
events                      printk_formats              snapshot            trace_stat
root@ThinkPad-T420:/sys/kernel/debug/tracing#
```

Function tracer

```
root@ThinkPad-T420:/sys/kernel/debug/tracing# cat tracing_on
1
root@ThinkPad-T420:/sys/kernel/debug/tracing# echo 0 > tracing_on
root@ThinkPad-T420:/sys/kernel/debug/tracing# echo function > current_tracer
root@ThinkPad-T420:/sys/kernel/debug/tracing# head -20 trace_pipe
^C
root@ThinkPad-T420:/sys/kernel/debug/tracing# echo 1 > tracing_on
root@ThinkPad-T420:/sys/kernel/debug/tracing# head -20 trace_pipe
head: cannot reposition file pointer for 'trace_pipe': Illegal seek
CPU:1 [LOST 37845 EVENTS]
          <idle>-0     [001] ....  3550.764030: _raw_spin_lock_irqsave <-__schedule
          <idle>-0     [001] d...  3550.764030: post_schedule_idle <-__schedule
          <idle>-0     [001] d...  3550.764031: idle_enter_fair <-post_schedule_idle
          <idle>-0     [001] d...  3550.764031: _raw_spin_unlock_irqrestore <-__schedule
          <idle>-0     [001] ....  3550.764031: tick_nohz_idle_enter <-cpu_startup_entry
          <idle>-0     [001] ....  3550.764031: set_cpu_sd_state_idle <-tick_nohz_idle_enter
          <idle>-0     [001] d...  3550.764032: __tick_nohz_idle_enter <-tick_nohz_idle_enter
          <idle>-0     [001] d...  3550.764032: ktime_get <-__tick_nohz_idle_enter
          <idle>-0     [001] d...  3550.764032: tick_nohz_stop_sched_tick <-__tick_nohz_idle_enter
          <idle>-0     [001] d...  3550.764032: timekeeping_max_deferment <-tick_nohz_stop_sched_tick
          <idle>-0     [001] d...  3550.764033: rcu_needs_cpu <-tick_nohz_stop_sched_tick
          <idle>-0     [001] d...  3550.764033: rcu_cpu_has_callbacks <-rcu_needs_cpu
          <idle>-0     [001] d...  3550.764033: get_next_timer_interrupt <-tick_nohz_stop_sched_tick
          <idle>-0     [001] d...  3550.764033: _raw_spin_lock <-get_next_timer_interrupt
          <idle>-0     [001] d...  3550.764034: _raw_spin_unlock <-get_next_timer_interrupt
          <idle>-0     [001] d...  3550.764034: hrtimer_get_next_event <-get_next_timer_interrupt
          <idle>-0     [001] d...  3550.764034: _raw_spin_lock_irqsave <-hrtimer_get_next_event
          <idle>-0     [001] d...  3550.764034: _raw_spin_unlock_irqrestore <-hrtimer_get_next_event
          <idle>-0     [001] d...  3550.764035: nohz_balance_enter_idle <-tick_nohz_stop_sched_tick
```

Function graph

```
root@ThinkPad-T420:/sys/kernel/debug/tracing# head -20 trace
# tracer: function_graph
#
# CPU  DURATION                  FUNCTION CALLS
# |     |   |                     |   |   |   |
 0)   3.664 us    |  } /* SyS_poll */
 0)               |  SyS_select() {
 0)               |    core_sys_select() {
 0)               |      do_select() {
 0)   0.040 us    |        fget_light();
 0)               |        sock_poll() {
 0)               |          unix_poll() {
 0)               |            __pollwait() {
 0)               |              add_wait_queue() {
 0)   0.044 us    |                _raw_spin_lock_irqsave();
 0)   0.058 us    |                _raw_spin_unlock_irqrestore();
 0)   0.833 us    |              }
 0)   1.256 us    |            }
 0)   1.718 us    |          }
 0)   2.083 us    |        }
 0)   0.044 us    |        fget_light();
```

sched_wakeup (available_events)

```
root@ThinkPad-T420:/sys/kernel/debug/tracing# echo 0 > tracing_on
root@ThinkPad-T420:/sys/kernel/debug/tracing# echo sched_wakeup > set_event
root@ThinkPad-T420:/sys/kernel/debug/tracing# echo 1 > tracing_on
root@ThinkPad-T420:/sys/kernel/debug/tracing# head -20 trace
# tracer: nop
#
# entries-in-buffer/entries-written: 90909/101305   #P:4
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
         rcuos/0-8     [003] d.h.  5724.583829: sched_wakeup: comm=rcu_sched pid=7 prio=120 success=1 target_cpu=003
            Xorg-1309  [002] d...  5724.583863: sched_wakeup: comm=compiz pid=2173 prio=120 success=1 target_cpu=003
            Xorg-1309  [002] d...  5724.584074: sched_wakeup: comm=xterm pid=2985 prio=120 success=1 target_cpu=001
            Xorg-1309  [002] d...  5724.584213: sched_wakeup: comm=compiz pid=2173 prio=120 success=1 target_cpu=003
            Xorg-1309  [002] d...  5724.584239: sched_wakeup: comm=ibus-x11 pid=1965 prio=120 success=1 target_cpu=002
        ibus-x11-1965  [002] d...  5724.584377: sched_wakeup: comm=Xorg pid=1309 prio=120 success=1 target_cpu=003
            Xorg-1309  [003] d...  5724.584463: sched_wakeup: comm=compiz pid=2173 prio=120 success=1 target_cpu=002
            Xorg-1309  [003] dN..  5724.584480: sched_wakeup: comm=ibus-x11 pid=1965 prio=120 success=1 target_cpu=003
            Xorg-1309  [003] dN..  5724.584524: sched_wakeup: comm=ibus-x11 pid=1965 prio=120 success=1 target_cpu=003
```

Latency tracers

- irqsoff
- preemptoff
- preemptirqsoff

irqoff

```
 # cat latency_trace
# tracer: irqsoff
#
irqsoff latency trace v1.1.5 on 2.6.26
--------------------------------------------------------------------
 latency: 12 us, #3/3, CPU#1 | (M:preempt VP:0, KP:0, SP:0 HP:0 #P:2)
    -----------------
    | task: bash-3730 (uid:0 nice:0 policy:0 rt_prio:0)
    -----------------
 => started at: sys_setpgid
 => ended at:   sys_setpgid

#                  _------=> CPU#
#                 / _-----=> irqs-off
#                | / _----=> need-resched
#                || / _---=> hardirq/softirq
#                ||| / _--=> preempt-depth
#                |||| /
#                |||||     delay
#  cmd     pid ||||| time  |   caller
#     \   /    |||||   \   |   /
    bash-3730  1d...    0us : _write_lock_irq (sys_setpgid)
    bash-3730  1d..1    1us+: _write_unlock_irq (sys_setpgid)
    bash-3730  1d..2   14us : trace_hardirqs_on (sys_setpgid)
```

preemptoff

```
 # cat latency_trace
# tracer: preemptoff
#
preemptoff latency trace v1.1.5 on 2.6.26-rc8
--------------------------------------------------------------------
 latency: 29 us, #3/3, CPU#0 | (M:preempt VP:0, KP:0, SP:0 HP:0 #P:2)
    -----------------
    | task: sshd-4261 (uid:0 nice:0 policy:0 rt_prio:0)
    -----------------
 => started at: do_IRQ
 => ended at:   __do_softirq

#                  _------=> CPU#
#                 / _-----=> irqs-off
#                | / _----=> need-resched
#                || / _---=> hardirq/softirq
#                ||| / _--=> preempt-depth
#                |||| /
#                |||||     delay
#  cmd     pid ||||| time  |   caller
#     \   /    |||||   \   |   /
    sshd-4261  0d.h.    0us+: irq_enter (do_IRQ)
    sshd-4261  0d.s.   29us : _local_bh_enable (__do_softirq)
    sshd-4261  0d.s1   30us : trace_preempt_on (__do_softirq)
```

preemptirqsoff

```
 # cat latency_trace
# tracer: preemptirqsoff
#
preemptirqsoff latency trace v1.1.5 on 2.6.26-rc8
--------------------------------------------------------------------
 latency: 293 us, #3/3, CPU#0 | (M:preempt VP:0, KP:0, SP:0 HP:0 #P:2)
    -----------------
    | task: ls-4860 (uid:0 nice:0 policy:0 rt_prio:0)
    -----------------
 => started at: apic_timer_interrupt
 => ended at:   __do_softirq

#                  _------=> CPU#
#                 / _-----=> irqs-off
#                | / _----=> need-resched
#                || / _---=> hardirq/softirq
#                ||| / _--=> preempt-depth
#                |||| /
#                |||||     delay
#  cmd     pid ||||| time  |   caller
#     \   /    |||||   \   |   /
      ls-4860  0d...    0us!: trace_hardirqs_off_thunk (apic_timer_interrupt)
      ls-4860  0d.s.  294us : _local_bh_enable (__do_softirq)
      ls-4860  0d.s1  294us : trace_preempt_on (__do_softirq)
```

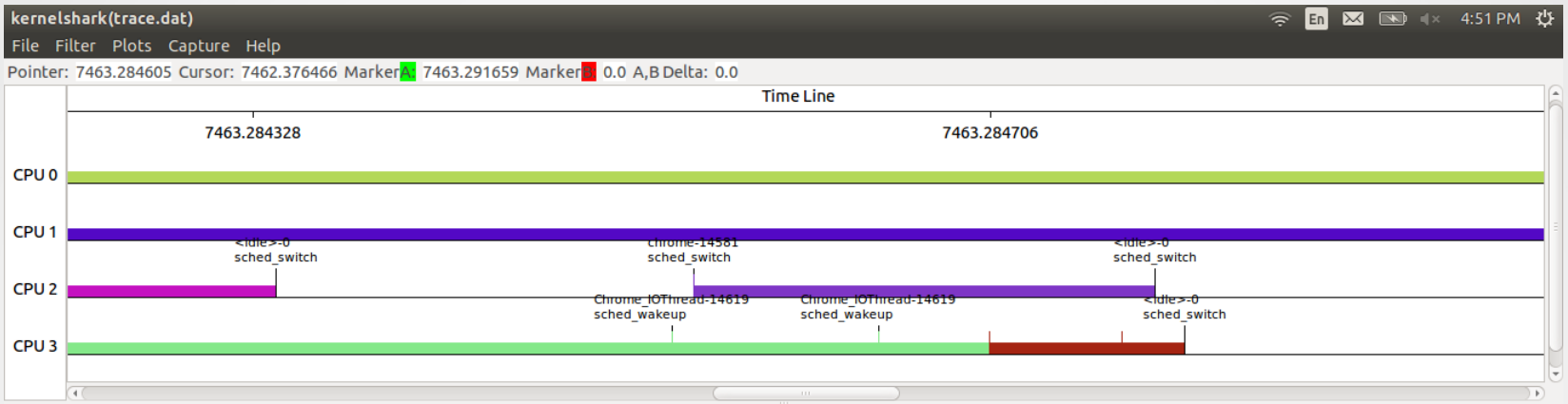
Can it be simple ?

- trace-cmd : Front end for ftrace
- trace-cmd record -e sched_wakup -e sched_switch
- trace-cmd record -p function_graph
- trace-cmd list

Demo

trace-cmd demo

KernelShark



Page 1 Search: Column: # contains graph follows

#	CPU	Time Stamp	Task	PID	Latency	Event	Info
0	2	7462.376424	trace-cmd	17892	dN..	sched_wakeup	kworker/2:0:17273 [120] success=1 CPU:002
1	2	7462.376432	trace-cmd	17892	d...	sched_switch	trace-cmd:17892 [120] S ==> kworker/2:0:17273 [120]
2	2	7462.376445	kworker/2:0	17273	d...	sched_wakeup	xterm:2985 [120] success=1 CPU:003
3	2	7462.376453	kworker/2:0	17273	d...	sched_switch	kworker/2:0:17273 [120] S ==> trace-cmd:17892 [120]
4	3	7462.376454	<idle>	0	d...	sched_switch	swapper/3:0 [120] R ==> xterm:2985 [120]
5	2	7462.376460	trace-cmd	17892	dN..	sched_wakeup	kworker/2:0:17273 [120] success=1 CPU:002
6	2	7462.376466	trace-cmd	17892	d...	sched_switch	trace-cmd:17892 [120] R ==> kworker/2:0:17273 [120]
7	2	7462.376472	kworker/2:0	17273	d...	sched_switch	kworker/2:0:17273 [120] S ==> trace-cmd:17892 [120]
8	2	7462.376500	trace-cmd	17892	d...	sched_switch	trace-cmd:17892 [120] S ==> swapper/2:0 [120]
9	1	7462.376507	trace-cmd	17893	d...	sched_wakeup	rcuos/1:9 [120] success=1 CPU:003
10	3	7462.376514	xterm	2985	d...	sched_switch	xterm:2985 [120] R ==> rcuos/1:9 [120]
11	3	7462.376522	rcuos/1	9	d...	sched_switch	rcuos/1:9 [120] S ==> xterm:2985 [120]
12	3	7462.376625	xterm	2985	d...	sched_wakeup	Xorg:1309 [120] success=1 CPU:002
13	2	7462.376634	<idle>	0	d...	sched_switch	swapper/2:0 [120] R ==> Xorg:1309 [120]

trace_marker

- Very useful to sync between userspace processes and kernel events
- Process enters some section having issues …
- Write trace marker
- Get through the section
- Write another end marker
- Get report in kernelshark

Persistent tracer

- Same concept as RAM console
- With tracer plugin hooks events/functions logged in ram
- Available after sudden reboot if RAM stays hot
- Would be great with kexec-tools

Perf

- Another performance tool
- Works on CPU performance counters
- Cache events
- Gpu performance
- Scheduling events
- Memory access
- CPU frequency and ticks
- irqs
- And many more

Debugging crashes during porting/boot

- Getting to the shell
- Disabling drivers
- Disabling smp
- Early printks
- jtag

Kernel Hacking Options

- Magic SysRq
- Slub debugging (under memory debugging)
- Lockup(spin_lock, mutexes) and hangs debugging
- Scheduler stats
- Early printk

Putting it all together

- Print debugging
- Control, flexibility info with procfs & sysfs
- /dev/mem
- Oops debugging
- RAM console, memory fencing, HW support
- Kexec, kdump, kgdb
- Performance debugging with ftrace, perf
- Kernel hacking options

Thank you !