Operating Systems Tutorial/Lab CW1

Semester 2 Academic year 24-25

Antonio Barbalace

abarbala@ed.ac.uk

Today

- Know your threads and processes
  - Linux's struct task_struct
  - gdb
- Tracing
  - Ftrace*
  - trace-cmd

Some material from: https://linux-kernel-labs.github.io/refs/heads/master/lectures/processes.html

`struct task_struct` (1/2)

- `struct task_struct` in Linux is (born as) the **Process Control Block (PCB)**
  - Where in the code?
  - Include/linux/sched.h (from line 737 to line 1546)

```
737 struct task_struct {
738 #ifdef CONFIG_THREAD_INFO_IN_TASK
739         /*
740          * For reasons of header soup (see current_thread_info()), this
741          * must be the first element of task_struct.
742          */
743         struct thread_info              thread_info;
744 #endif
745         unsigned int                    __state;
746
747 #ifdef CONFIG_PREEMPT_RT
748         /* saved state for "spinlock sleepers" */
749         unsigned int                    saved_state;
750 #endif
751
752         /*
753          * This begins the randomizable portion of task_struct. Only
754          * scheduling-critical items should be added above here.
755          */
756         randomized_struct_fields_start
757
758         void                            *stack;
759         refcount_t                      usage;
760         /* Per task flags (PF_*), defined further below: */
761         unsigned int                    flags;
762         unsigned int                    ptrace;
763
```

`struct task_struct` (2/2)

- `struct task_struct` in Linux is (born as) the **Process Control Block (PCB)**
  - Where in the code?
  - Include/linux/sched.h (from line 737 to line 1546)

```
$ pahole -C task_struct vmlinux

struct task_struct {
    struct thread_info thread_info;                 /*     0     8 */
    volatile long int          state;               /*     8     4 */
    void *                     stack;               /*    12     4 */

    ...

    /* --- cacheline 45 boundary (2880 bytes) --- */
    struct thread_struct thread __attribute__((__aligned__(64))); /*  2880  4288 */

    /* size: 7168, cachelines: 112, members: 155 */
    /* sum members: 7148, holes: 2, sum holes: 12 */
    /* sum bitfield members: 7 bits, bit holes: 2, sum bit holes: 57 bits */
    /* paddings: 1, sum paddings: 2 */
    /* forced alignments: 6, forced holes: 2, sum forced holes: 12 */
} __attribute__((__aligned__(64)));
```

Getting into know your Process(es) (1/2)

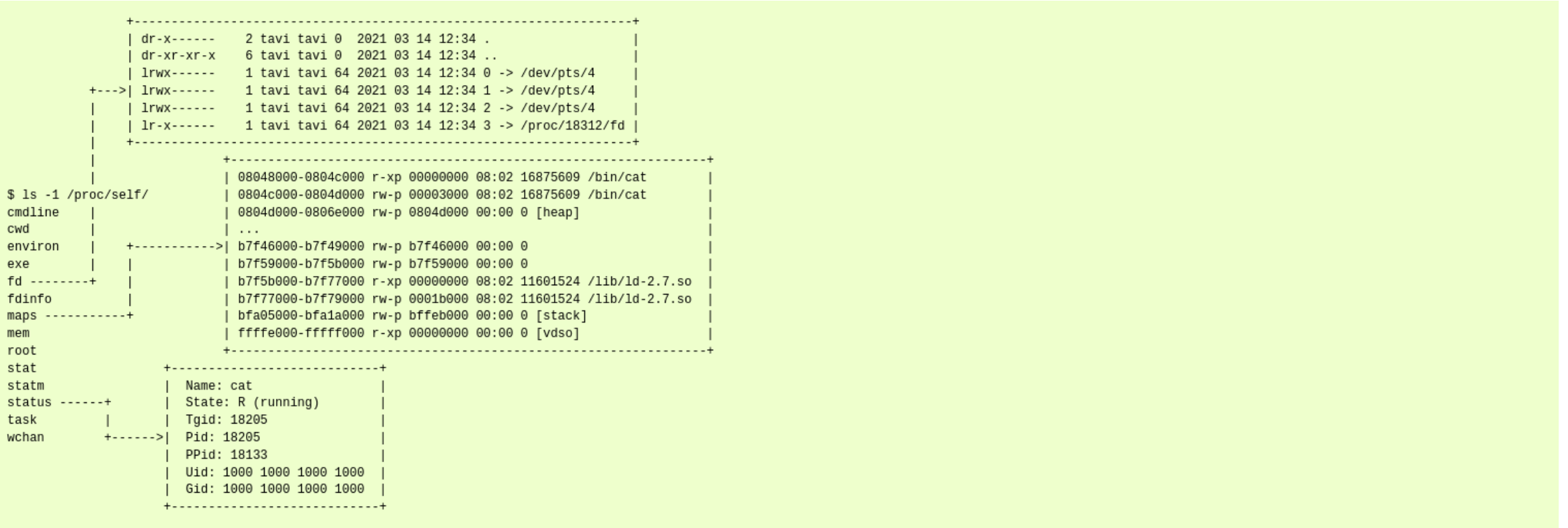
- In Linux (UNIX) a summary of the resources of a process obtained
  - From `/proc/<pid>`
  - Where `<pid>` is the process id of the process, or `self`

```
[hendry]abarbala: ls --color /proc/self
arch_status  coredump_filter     gid_map    mounts         pagemap      setgroups     task
attr         cpu_resctrl_groups  io         mountstats     patch_state  smaps         timens_offsets
autogroup    cpuset              limits     net            personality  smaps_rollup  timers
auxv         cwd                 loginuid   ns             projid_map   stack         timerslack_ns
cgroup       environ             map_files  numa_maps      root         stat          uid_map
clear_refs   exe                 maps       oom_adj        sched        statm         wchan
cmdline      fd                  mem        oom_score      schedstat    status
comm         fdinfo              mountinfo  oom_score_adj  sessionid    syscall
```

This is on student.compute or staff.compute, try the command also in your VM.

Getting into know your Process(es) (2/2)

- In Linux (UNIX) a summary of the resources of a process obtained
  - From `/proc/<pid>`
  - Where `<pid>` is the process id of the process, or `self`

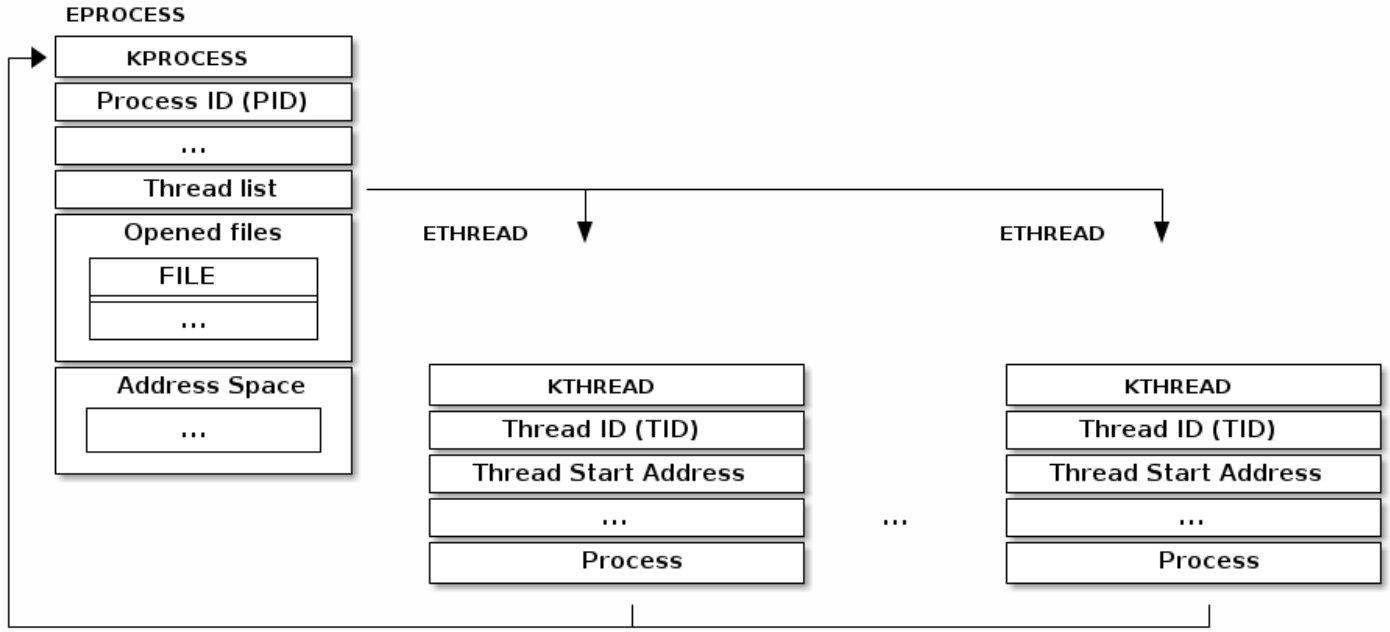


What about Threads?

- A thread is the **basic unit** the Linux kernel scheduler uses to allow applications to run on CPU
  - Each thread has its own stack and register values (thread execution state)
  - A thread runs in the context of a process
    - All threads in the same process share OS resources
  - The kernel schedules threads, not processes
    - User-level threads (e.g. fibers, coroutines, etc.) are not visible at the kernel level

TCB in Linux? (1/2)

- The typical thread implementation is one where the threads is implemented as a separate data structure
  - Then linked to the process data structure
  - For example, the Windows kernel uses such an implementation

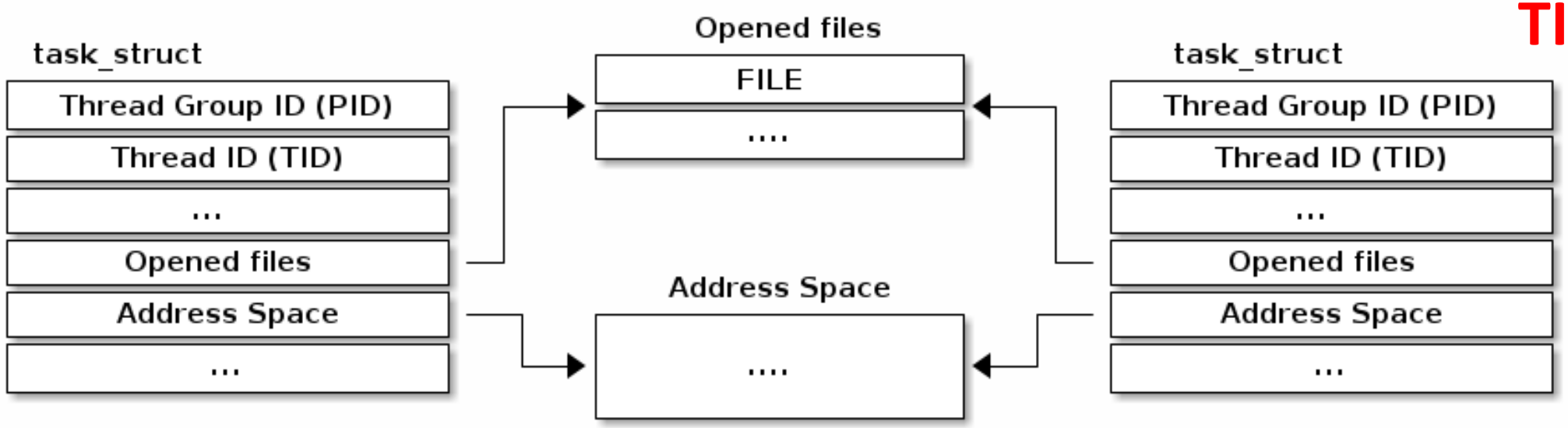


TCB in Linux? (2/2)

- Linux uses a different implementation
  - The basic unit is the **task** (`struct task_struct`)
  - It is used for both **threads** and **processes**
  - Instead of embedding OS resources in the task descriptor it has pointers to these resources

PID is tgid

TID is pid



Creating Threads or Processes

- In Linux, a thread or process is created with `clone()`
  - `fork()` calls `clone()`
  - `pthread_create()` calls `clone()`

- It allows the caller to decide **what resources be shared** with the parent, what should be **copied** or **isolated**
  - `CLONE_FILES` - shares the file descriptor table with the parent
  - `CLONE_VM` - shares the address space with the parent
  - `CLONE_FS` - shares the filesystem information (root directory, current directory) with the parent
  - `CLONE_NEWNS` - does not share the mount namespace with the parent
  - `CLONE_NEWIPC` - does not share the IPC namespace with the parent
    - System V IPC objects, POSIX message queues
  - `CLONE_NEWNET` - does not share the networking namespaces with the parent
    - network interfaces, routing table

Remember

- Whom is running now? (On the CPU where the code is executing)
  - https://elixir.bootlin.com/linux/v6.1.75/source/include/asm-generic/current.h#L8

/ include / asm-generic / current.h

```
1   /* SPDX-License-Identifier: GPL-2.0 */
2   #ifndef __ASM_GENERIC_CURRENT_H
3   #define __ASM_GENERIC_CURRENT_H
4
5   #include <linux/thread_info.h>
6
7   #define get_current() (current_thread_info()->task)
8   #define current get_current()
9
10  #endif /* __ASM_GENERIC_CURRENT_H */
```

What about GDB?

- XRDP Connection
  - On Windows use “Remote desktop Connection” application
  - `sXXXXXXX.remote.inf.ed.ac.uk`

- Open a shell and connect to student.compute (or other machines)
  - `sXXXXXXX: ssh student.compute`

- **Start QEMU** first
  - `sXXXXXXX: qemu-system-x86_64 -m 4G -smp 4 -drive file=/disk/scratch/sXXXXXXX/debian.qcow2 -nographic -S -gdb tcp::YYYYY -kernel /disk/scratch/sXXXXXXX/linux-6.1.75/arch/x86_64/boot/bzImage -append “root=/dev/sda1 console=ttyS0 earlyprintk=ttyS0 nokaslr”`

- **Start gdb** second – you need another console/terminal
  - `sXXXXXXX: cd /disk/scratch/sXXXXXXX/linux-6.1.75`
  - `sXXXXXXX: gdb vmlinux`
  - `(gdb) target remote localhost:YYYYY`

This is the kernel we compiled before !!!

Who is running **now**? (1/2)

- Let's add a **breakpoint**
  - `(gdb) break get_current`
  - `(gdb) c`
  - `(gdb) layout next`
  - `(gdb) layout next`
  - `(gdb) layout next`
  - `(gdb) si`
    - `asm: mov %gs:0x1ad00,%rdx`
- Now, the **magic**
  - What are the **TID and PID?**
  - `(gdb) print ((struct task_struct*)($rdx))->tgid`
  - `(gdb) print ((struct task_struct*)($rdx))->pid`
  - What is the process **name?**
  - `(gdb) print ((struct task_struct*)($rdx))->comm`

Who is running **now**? (2/2)

- Let's add a **breakpoint**
  - (gdb) break get_current
  - (gdb) c
  - (gdb) layout next
  - (gdb) layout next
  - (gdb) layout next
  - (gdb) si
    - asm: mov %gs:0x1ad00,%rdx
- Now, the **magic**
  - What are the **TID and PID?**
  - (gdb) print ((struct task_struct*)(
  - (gdb) print ((struct task_struct*)(
  - What is the process **name?**
  - (gdb) print ((struct task_struct*)(

```
File  Edit  View  Search  Terminal  Help
./include/linux/sched.h
1913        extern struct task_struct *idle_task(int cpu);
1914
1915        /**
1916         * is_idle_task - is the specified task an idle task?
1917         * @p: the task in question.
1918         *
1919         * Return: 1 if @p is an idle task. 0 otherwise.
1920         */
1921        static __always_inline bool is_idle_task(const struct task_struct *p)
1922        {
>1923                return !!(p->flags & PF_IDLE);
1924        }
1925
1926        extern struct task_struct *curr_task(int cpu);
1927        extern void ia64_set_curr_task(int cpu, struct task_struct *p);
1928
1929        void yield(void);
1930
1931        union thread_union {
1932        #ifndef CONFIG_ARCH_TASK_STRUCT_ON_STACK
1933                struct task_struct task;
1934        #endif
```

```
   0xffffffff81d0f780 <irqentry_enter>     testb  $0x3,0x88(%rdi)
   0xffffffff81d0f787 <irqentry_enter+7>   jne    0xffffffff81d0f79f <irqentry_enter+31>
   0xffffffff81d0f789 <irqentry_enter+9>   xor    %eax,%eax
B+ 0xffffffff81d0f78b <irqentry_enter+11>  mov    %gs:0x1ad00,%rdx
  >0xffffffff81d0f794 <irqentry_enter+20>  testb  $0x2,0x2c(%rdx)
   0xffffffff81d0f798 <irqentry_enter+24>  jne    0xffffffff81d0f7ab <irqentry_enter+43>
   0xffffffff81d0f79a <irqentry_enter+26>  retq
   0xffffffff81d0f79b <irqentry_enter+27>  int3
   0xffffffff81d0f79c <irqentry_enter+28>  int3
   0xffffffff81d0f79d <irqentry_enter+29>  int3
   0xffffffff81d0f79e <irqentry_enter+30>  int3
   0xffffffff81d0f79f <irqentry_enter+31>  callq  0xffffffff81d0f770 <irqentry_enter_from_user_mode>
   0xffffffff81d0f7a4 <irqentry_enter+36>  xor    %eax,%eax
   0xffffffff81d0f7a6 <irqentry_enter+38>  retq
   0xffffffff81d0f7a7 <irqentry_enter+39>  int3
   0xffffffff81d0f7a8 <irqentry_enter+40>  int3
   0xffffffff81d0f7a9 <irqentry_enter+41>  int3
   0xffffffff81d0f7aa <irqentry_enter+42>  int3
   0xffffffff81d0f7ab <irqentry_enter+43>  callq  0xffffffff81d0fc60 <ct_irq_enter>
   0xffffffff81d0f7b0 <irqentry_enter+48>  mov    $0x1,%eax
   0xffffffff81d0f7b5 <irqentry_enter+53>  retq
   0xffffffff81d0f7b6 <irqentry_enter+54>  int3
```

```
remote Thread 1.3 In: irqentry_enter                                L1923 PC: 0xffffffff81d0f79
(gdb) layout next
(gdb) layout next
(gdb) b get_current
Breakpoint 1 at 0xffffffff81001a70: get_current. (3757 locations)
(gdb) c
Continuing.
[Switching to Thread 1.3]

Thread 3 hit Breakpoint 1, 0xffffffff81d0f78b in get_current () at ./arch/x86/include/asm/current.h:15
(gdb) si
irqentry_enter (regs=regs@entry=0xffffc9000008be38) at ./include/linux/sched.h:1923
(gdb) focus cmd
Focus set to cmd window.
(gdb) print ((struct task_struct*)($rdx))->tgid
$1 = 0
(gdb) print ((struct task_struct*)($rdx))->pid
$2 = 0
(gdb) print ((struct task_struct*)($rdx))->comm
$3 = "swapper/2\000\000\000\000\000\000"
(gdb)
```

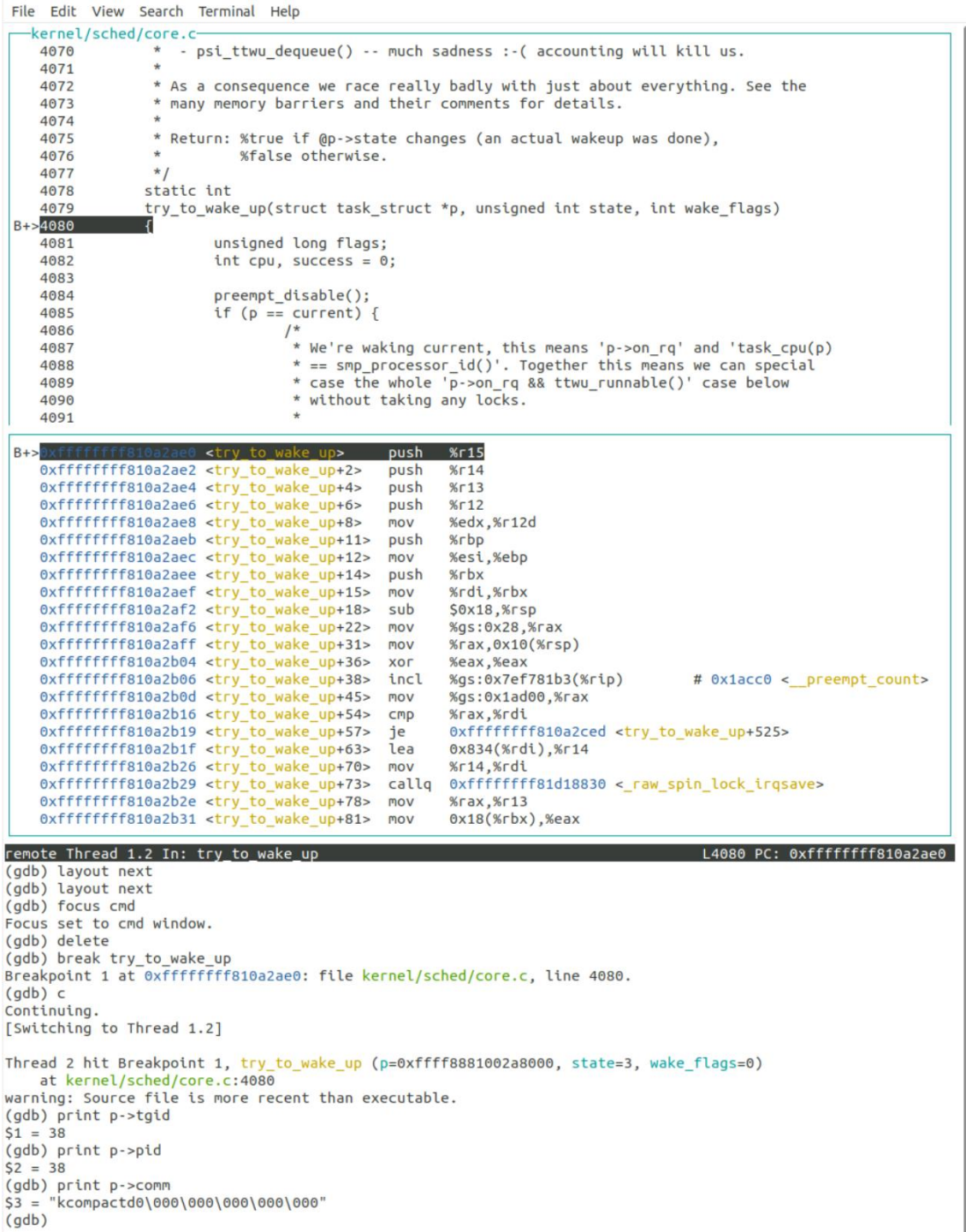
Who is **next**? (1/2)

- Let's add another **breakpoint**
  - Clean breakpoints first
  - `(gdb) delete`
  - Add another breakpoint
  - `(gdb) break try_to_wake_up`

- Now, a bit less **magic**
  - What are the **TID and PID?**
  - `(gdb) print p->tgid`
  - `(gdb) print p->pid`
  - What is the process **name?**
  - `(gdb) print p->comm`

Who is **next**? (2/2)

- Let's add another **breakpoint**
  - Clean breakpoints first
  - `(gdb) delete`
  - Add another breakpoint
  - `(gdb) break try_to_wake_up`
- Now, a bit less **magic**
  - What are the **TID and PID?**
  - `(gdb) print p->tgid`
  - `(gdb) print p->pid`
  - What is the process **name?**
  - `(gdb) print p->comm`



Who is next **live!!!**

- Let's add a **command** **to a breakpoint**
  - Clean breakpoints first
  - `(gdb) delete`
  - Read the breakpoint
  - `(gdb) break try_to_wake_up`
  - `(gdb) command`
  - `> print p->comm`
  - `> cont`
  - `> end`
  - `(gdb) c`
- Enjoy the flow of **context switches!!!**
- Stop with **CTRL-c**

Linux and GDB Integration

- Linux provides scripts for **gdb**, but oops ...
  - **You may need to fix the scripts before loading them!!!**
    - `sXXXXXXX: cd /disk/scratch/sXXXXXXX/linux-6.1.75`
    - `sXXXXXXX: sed -i 's/1UL/0x01/g' scripts/gdb/linux/constants.py`

- Let's load such (Python) scripts in **gdb**
  - **This assumes you already have a VM running**
  - `sXXXXXXX: cd /disk/scratch/sXXXXXXX/linux-6.1.75`
  - `sXXXXXXX: gdb vmlinux`
  - `(gdb) target remote localhost:YYYYY`
  - `(gdb) c`
  - CTRL-c
  - `(gdb) source vmlinux-gdb.py`
  - `(gdb) lx-` TAB TAB

```
(gdb) source vmlinux-gdb.py
(gdb) lx-
lx-clk-summary        lx-device-list-class  lx-iomem              lx-ps
lx-cmdline            lx-device-list-tree   lx-ioports            lx-symbols
lx-configdump         lx-dmesg              lx-list-check         lx-timerlist
lx-cpus               lx-fdtdump            lx-lsmod              lx-version
lx-device-list-bus    lx-genpd-summary      lx-mounts
(gdb) lx-
```

https://docs.kernel.org/dev-tools/gdb-kernel-debugging.html

Linux and GDB Integration …

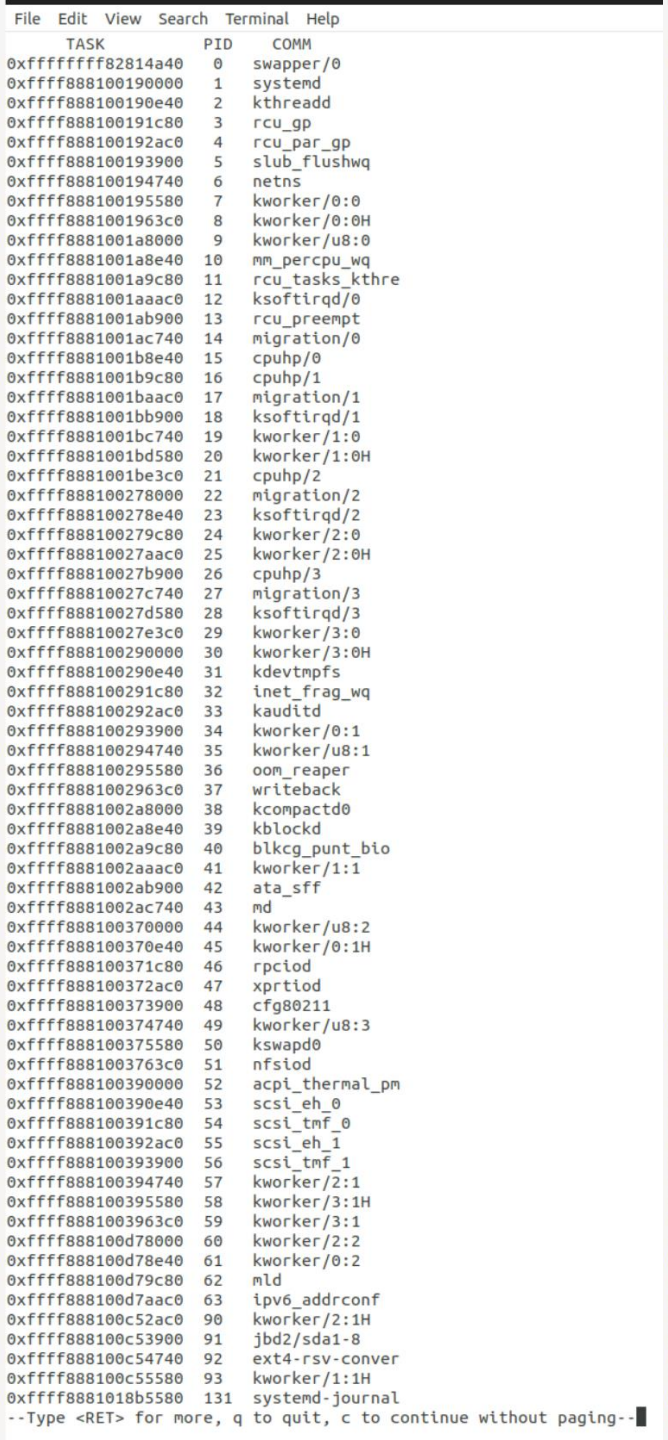
- The **scripts** include Linux's kernel
  - Commands
  - Functions
- To list them all
  - `(gdb) apropos lx`

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

https://docs.kernel.org/dev-tools/gdb-kernel-debugging.html

lx-ps

- Like ps in user-space
- Lists the current processes



```
File  Edit  View  Search  Terminal  Help
      TASK          PID    COMM
0xffffffff82814a40    0   swapper/0
0xffff888100190000    1   systemd
0xffff888100190e40    2   kthreadd
0xffff888100191c80    3   rcu_gp
0xffff888100192ac0    4   rcu_par_gp
0xffff888100193900    5   slub_flushwq
0xffff888100194740    6   netns
0xffff888100195580    7   kworker/0:0
0xffff8881001963c0    8   kworker/0:0H
0xffff8881001a8000    9   kworker/u8:0
0xffff8881001a8e40   10   mm_percpu_wq
0xffff8881001a9c80   11   rcu_tasks_kthre
0xffff8881001aaac0   12   ksoftirqd/0
0xffff8881001ab900   13   rcu_preempt
0xffff8881001ac740   14   migration/0
0xffff8881001b8e40   15   cpuhp/0
0xffff8881001b9c80   16   cpuhp/1
0xffff8881001baac0   17   migration/1
0xffff8881001bb900   18   ksoftirqd/1
0xffff8881001bc740   19   kworker/1:0
0xffff8881001bd580   20   kworker/1:0H
0xffff8881001be3c0   21   cpuhp/2
0xffff888100278000   22   migration/2
0xffff888100278e40   23   ksoftirqd/2
0xffff888100279c80   24   kworker/2:0
0xffff88810027aac0   25   kworker/2:0H
0xffff88810027b900   26   cpuhp/3
0xffff88810027c740   27   migration/3
0xffff88810027d580   28   ksoftirqd/3
0xffff88810027e3c0   29   kworker/3:0
0xffff888100290000   30   kworker/3:0H
0xffff888100290e40   31   kdevtmpfs
0xffff888100291c80   32   inet_frag_wq
0xffff888100292ac0   33   kauditd
0xffff888100293900   34   kworker/0:1
0xffff888100294740   35   kworker/u8:1
0xffff888100295580   36   oom_reaper
0xffff8881002963c0   37   writeback
0xffff8881002a8000   38   kcompactd0
0xffff8881002a8e40   39   kblockd
0xffff8881002a9c80   40   blkcg_punt_bio
0xffff8881002aaac0   41   kworker/1:1
0xffff8881002ab900   42   ata_sff
0xffff8881002ac740   43   md
0xffff888100370000   44   kworker/u8:2
0xffff888100370e40   45   kworker/0:1H
0xffff888100371c80   46   rpciod
0xffff888100372ac0   47   xprtiod
0xffff888100373900   48   cfg80211
0xffff888100374740   49   kworker/u8:3
0xffff888100375580   50   kswapd0
0xffff8881003763c0   51   nfsiod
0xffff888100390000   52   acpi_thermal_pm
0xffff888100390e40   53   scsi_eh_0
0xffff888100391c80   54   scsi_tmf_0
0xffff888100392ac0   55   scsi_eh_1
0xffff888100393900   56   scsi_tmf_1
0xffff888100394740   57   kworker/2:1
0xffff888100395580   58   kworker/3:1H
0xffff8881003963c0   59   kworker/3:1
0xffff888100d78000   60   kworker/2:2
0xffff888100d78e40   61   kworker/0:2
0xffff888100d79c80   62   mld
0xffff888100d7aac0   63   ipv6_addrconf
0xffff888100c52ac0   90   kworker/2:1H
0xffff888100c53900   91   jbd2/sda1-8
0xffff888100c54740   92   ext4-rsv-conver
0xffff888100c55580   93   kworker/1:1H
0xffff8881018b5580  131   systemd-journal
--Type <RET> for more, q to quit, c to continue without paging--
```

```
File  Edit  View  Search  Terminal  Help
0xffff8881001b9c80   16   cpuhp/1
0xffff8881001baac0   17   migration/1
0xffff8881001bb900   18   ksoftirqd/1
0xffff8881001bc740   19   kworker/1:0
0xffff8881001bd580   20   kworker/1:0H
0xffff8881001be3c0   21   cpuhp/2
0xffff888100278000   22   migration/2
0xffff888100278e40   23   ksoftirqd/2
0xffff888100279c80   24   kworker/2:0
0xffff88810027aac0   25   kworker/2:0H
0xffff88810027b900   26   cpuhp/3
0xffff88810027c740   27   migration/3
0xffff88810027d580   28   ksoftirqd/3
0xffff88810027e3c0   29   kworker/3:0
0xffff888100290000   30   kworker/3:0H
0xffff888100290e40   31   kdevtmpfs
0xffff888100291c80   32   inet_frag_wq
0xffff888100292ac0   33   kauditd
0xffff888100293900   34   kworker/0:1
0xffff888100294740   35   kworker/u8:1
0xffff888100295580   36   oom_reaper
0xffff8881002963c0   37   writeback
0xffff8881002a8000   38   kcompactd0
0xffff8881002a8e40   39   kblockd
0xffff8881002a9c80   40   blkcg_punt_bio
0xffff8881002aaac0   41   kworker/1:1
0xffff8881002ab900   42   ata_sff
0xffff8881002ac740   43   md
0xffff888100370000   44   kworker/u8:2
0xffff888100370e40   45   kworker/0:1H
0xffff888100371c80   46   rpciod
0xffff888100372ac0   47   xprtiod
0xffff888100373900   48   cfg80211
0xffff888100374740   49   kworker/u8:3
0xffff888100375580   50   kswapd0
0xffff8881003763c0   51   nfsiod
0xffff888100390000   52   acpi_thermal_pm
0xffff888100390e40   53   scsi_eh_0
0xffff888100391c80   54   scsi_tmf_0
0xffff888100392ac0   55   scsi_eh_1
0xffff888100393900   56   scsi_tmf_1
0xffff888100394740   57   kworker/2:1
0xffff888100395580   58   kworker/3:1H
0xffff8881003963c0   59   kworker/3:1
0xffff888100d78000   60   kworker/2:2
0xffff888100d78e40   61   kworker/0:2
0xffff888100d79c80   62   mld
0xffff888100d7aac0   63   ipv6_addrconf
0xffff888100c52ac0   90   kworker/2:1H
0xffff888100c53900   91   jbd2/sda1-8
0xffff888100c54740   92   ext4-rsv-conver
0xffff888100c55580   93   kworker/1:1H
0xffff8881018b5580  131   systemd-journal
--Type <RET> for more, q to quit, c to continue without paging--
0xffff888100c50e40  138   kworker/3:2
0xffff888100c50000  154   kworker/3:3
0xffff8881018b3900  162   systemd-udevd
0xffff888100c563c0  163   kworker/1:2
0xffff888103b60000  165   systemd-network
0xffff8881018b1c80  210   systemd-resolve
0xffff8881018b4740  211   systemd-timesyn
0xffff888103b63900  213   dbus-daemon
0xffff888103b65580  215   systemd-logind
0xffff888103b663c0  217   agetty
0xffff888103b62ac0  218   login
0xffff888103b64740  222   unattended-upgr
0xffff888103b61c80  229   systemd
0xffff8881018b63c0  230   (sd-pam)
0xffff888103a78e40  235   bash
0xffff888103b60e40  246   kworker/2:3
(gdb)
```

lx_current()

- Like `get_current()`
- Returns the current task
  - A `struct task_struct` pointer

```
(gdb) p $lx_current().tgid
$11 = 0
(gdb) p $lx_current().pid
$12 = 0
(gdb) p $lx_current().comm
$13 = "swapper/0\000\000\000\000\000\000"
(gdb) p $lx_current()
```

```
$14 = {thread_info = {flags = 16384, syscall_work = 0, status = 0, cpu = 0}, __state = 0,
  stack = 0xffffffff82800000, usage = {refs = {counter = 2}}, flags = 69206018, ptrace = 0, on_cpu = 1,
  wake_entry = {llist = {next = 0x0 <fixed_percpu_data>}, {u_flags = 48, a_flags = {counter = 48}},
    src = 0, dst = 0}, wakee_flips = 4, wakee_flip_decay_ts = 4294785747, last_wakee = 0xffff8881001ab900,
  recent_used_cpu = 0, wake_cpu = 0, on_rq = 1, prio = 120, static_prio = 120, normal_prio = 120,
  rt_priority = 0, se = {load = {weight = 1048576, inv_weight = 4194304}, run_node = {
      __rb_parent_color = 0, rb_right = 0x0 <fixed_percpu_data>, rb_left = 0x0 <fixed_percpu_data>},
    group_node = {next = 0xffffffff82814ae8 <init_task+168>, prev = 0xffffffff82814ae8 <init_task+168>},
    on_rq = 0, exec_start = 388816297, sum_exec_runtime = 0, vruntime = 0, prev_sum_exec_runtime = 0,
    nr_migrations = 0, depth = 0, parent = 0x0 <fixed_percpu_data>, cfs_rq = 0xffff88813bc29e80,
    my_q = 0x0 <fixed_percpu_data>, runnable_weight = 0, avg = {last_update_time = 0, load_sum = 0,
      runnable_sum = 0, util_sum = 0, period_contrib = 0, load_avg = 0, runnable_avg = 0, util_avg = 0,
      util_est = {enqueued = 0, ewma = 0}}}, rt = {run_list = {next = 0xffffffff82814bc0 <init_task+384>,
      prev = 0xffffffff82814bc0 <init_task+384>}, timeout = 0, watchdog_stamp = 0, time_slice = 100,
    on_rq = 0, on_list = 0, back = 0x0 <fixed_percpu_data>}, dl = {rb_node = {
      __rb_parent_color = 18446744071604095984, rb_right = 0x0 <fixed_percpu_data>,
      rb_left = 0x0 <fixed_percpu_data>}, dl_runtime = 0, dl_deadline = 0, dl_period = 0, dl_bw = 0,
    dl_density = 0, runtime = 0, deadline = 0, flags = 0, dl_throttled = 0, dl_yielded = 0,
    dl_non_contending = 0, dl_overrun = 0, dl_timer = {node = {node = {
          __rb_parent_color = 18446744071604096072, rb_right = 0x0 <fixed_percpu_data>,
          rb_left = 0x0 <fixed_percpu_data>}, expires = 0}, _softexpires = 0,
      function = 0xffffffff810b44f0 <dl_task_timer>, base = 0xffff88813bc1df40, state = 0 '\000',
      is_rel = 0 '\000', is_soft = 0 '\000', is_hard = 1 '\001'}, inactive_timer = {node = {node = {
          __rb_parent_color = 18446744071604096136, rb_right = 0x0 <fixed_percpu_data>,
          rb_left = 0x0 <fixed_percpu_data>}, expires = 0}, _softexpires = 0,
      function = 0xffffffff810b1e10 <inactive_task_timer>, base = 0xffff88813bc1df40, state = 0 '\000',
      is_rel = 0 '\000', is_soft = 0 '\000', is_hard = 1 '\001'},
    pi_se = 0xffffffff82814bf0 <init_task+432>}, sched_class = 0xffffffff82670240 <idle_sched_class>,
  sched_task_group = 0xffffffff8320c800 <root_task_group>, stats = {wait_start = 0, wait_max = 0,
    wait_count = 0, wait_sum = 0, iowait_count = 0, iowait_sum = 0, sleep_start = 0, sleep_max = 0,
    sum_sleep_runtime = 0, block_start = 0, block_max = 0, sum_block_runtime = 0, exec_max = 0,
    slice_max = 0, nr_migrations_cold = 0, nr_failed_migrations_affine = 0,
    nr_failed_migrations_running = 0, nr_failed_migrations_hot = 0, nr_forced_migrations = 0,
    nr_wakeups = 0, nr_wakeups_sync = 0, nr_wakeups_migrate = 0, nr_wakeups_local = 0,
    nr_wakeups_remote = 0, nr_wakeups_affine = 0, nr_wakeups_affine_attempts = 0, nr_wakeups_passive = 0,
    nr_wakeups_idle = 0}, btrace_seq = 0, policy = 0, nr_cpus_allowed = 1,
  cpus_ptr = 0xffffffff82814e20 <init_task+992>, user_cpus_ptr = 0x0 <fixed_percpu_data>, cpus_mask = {
    bits = {1}}, migration_pending = 0x0 <fixed_percpu_data>, migration_disabled = 0, migration_flags = 0,
  rcu_read_lock_nesting = 0, rcu_read_unlock_special = {b = {blocked = 0 '\000', need_qs = 0 '\000',
      exp_hint = 0 '\000', need_mb = 0 '\000'}, s = 0}, rcu_node_entry = {
    next = 0xffffffff82814e40 <init_task+1024>, prev = 0xffffffff82814e40 <init_task+1024>},
  rcu_blocked_node = 0x0 <fixed_percpu_data>, rcu_tasks_nvcsw = 0, rcu_tasks_holdout = 0 '\000',
  rcu_tasks_idx = 0 '\000', rcu_tasks_idle_cpu = -1, rcu_tasks_holdout_list = {
    next = 0xffffffff82814e68 <init_task+1064>, prev = 0xffffffff82814e68 <init_task+1064>}, sched_info = {
    pcount = 0, run_delay = 0, last_arrival = 0, last_queued = 0}, tasks = {next = 0xffff888100190458,
    prev = 0xffff888103b61298}, pushable_tasks = {prio = 140, prio_list = {
      next = 0xffffffff82814eb0 <init_task+1136>, prev = 0xffffffff82814eb0 <init_task+1136>}, node_list = {
      next = 0xffffffff82814ec0 <init_task+1152>, prev = 0xffffffff82814ec0 <init_task+1152>}},
  pushable_dl_tasks = {__rb_parent_color = 0, rb_right = 0x0 <fixed_percpu_data>,
    rb_left = 0x0 <fixed_percpu_data>}, mm = 0x0 <fixed_percpu_data>, active_mm = 0xffff888100d8e800,
  rss_stat = {events = 0, count = {0, 0, 0, 0}}, exit_state = 0, exit_code = 0, exit_signal = 0,
  pdeath_signal = 0, jobctl = 0, personality = 0, sched_reset_on_fork = 0, sched_contributes_to_load = 0,
  sched_migrated = 0, sched_remote_wakeup = 0, in_execve = 0, in_iowait = 0, restore_sigmask = 0,
  no_cgroup_migration = 0, frozen = 0, use_memdelay = 0, in_eventfd = 0, reported_split_lock = 0,
  in_thrashing = 0, atomic_flags = 0, restart_block = {arch_data = 0,
    fn = 0xffffffff8107dc40 <do_no_restart_syscall>, {futex = {uaddr = 0x0 <fixed_percpu_data>, val = 0,
        flags = 0, bitset = 0, time = 0, uaddr2 = 0x0 <fixed_percpu_data>}, nanosleep = {clockid = 0,
        type = TT_NONE, {rmtp = 0x0 <fixed_percpu_data>, compat_rmtp = 0x0 <fixed_percpu_data>},
        expires = 0}, poll = {ufds = 0x0 <fixed_percpu_data>, nfds = 0, has_timeout = 0, tv_sec = 0,
        tv_nsec = 0}}}, pid = 0, tgid = 0, stack_canary = 16661359093833919744,
  real_parent = 0xffffffff82814a40 <init_task>, parent = 0xffffffff82814a40 <init_task>, children = {
    next = 0xffff888100190568, prev = 0xffff8881001913a8}, sibling = {
    next = 0xffffffff82814fa8 <init_task+1384>, prev = 0xffffffff82814fa8 <init_task+1384>},
  group_leader = 0xffffffff82814a40 <init_task>, ptraced = {next = 0xffffffff82814fc0 <init_task+1408>,
    prev = 0xffffffff82814fc0 <init_task+1408>}, ptrace_entry = {
    next = 0xffffffff82814fd0 <init_task+1424>, prev = 0xffffffff82814fd0 <init_task+1424>},
  thread_pid = 0xffffffff828516a0 <init_struct_pid>, pid_links = {{next = 0x0 <fixed_percpu_data>,
      pprev = 0x0 <fixed_percpu_data>}, {next = 0x0 <fixed_percpu_data>, pprev = 0x0 <fixed_percpu_data>}, {
      next = 0x0 <fixed_percpu_data>, pprev = 0x0 <fixed_percpu_data>}, {next = 0x0 <fixed_percpu_data>,
      pprev = 0x0 <fixed_percpu_data>}}, thread_group = {next = 0xffffffff82815028 <init_task+1512>,
--Type <RET> for more, q to quit, c to continue without paging--
```

Any other tool? **Tracing!**

- **Functional** tracers
  - ftrace
    - Kernel-space
  - trace-cmd
    - User-space tool
  - kernel-shark ***
    - GUI
  - Lttng ***
    - GUI
  - ...

kernel-shark https://lwn.net/Articles/425583/

trace-cmd

- ftrace is Linux function tracer
  - Not that simple to play with
  - trace-cmd is a suite of programs to interact with it in a simple way

- The following commands must be typed **INTO** the VM
  - Let's make sure the VM is connected to the network
    - # dhclient
    - # ping 1.1.1.1
  - If the ping works, continue
    - # apt-get install trace-cmd

trace-cmd https://lwn.net/Articles/410200/

trace-cmd **first example**

- Let's **record** a trace
  - `# trace-cmd record -e ext4 ls`

```
root@systems-nuts:~# trace-cmd record -e ext4 ls
trace.dat  trace.dat.cpu0  trace.dat.cpu1  trace.dat.cpu2  trace.dat.cpu3
CPU0 data recorded at offset=0x103000
    0 bytes in size (0 uncompressed)
CPU1 data recorded at offset=0x103000
    141 bytes in size (4096 uncompressed)
CPU2 data recorded at offset=0x104000
    0 bytes in size (0 uncompressed)
CPU3 data recorded at offset=0x104000
    0 bytes in size (0 uncompressed)
```

- Let's **have a look** at the trace
  - `#  trace-cmd report`

```
root@systems-nuts:~# trace-cmd report
cpus=4
             ls-4805  [001] 20224.187883: ext4_es_lookup_extent_enter: dev 8,1 ino 8200 lblk 0
             ls-4805  [001] 20224.187903: ext4_es_lookup_extent_exit: dev 8,1 ino 8200 found 1 [0/1) 81R
             ls-4805  [001] 20224.187999: ext4_journal_start:   dev 8,1 blocks 2, rsv_blocks 0, revoke_0
             ls-4805  [001] 20224.188018: ext4_mark_inode_dirty: dev 8,1 ino 8200 caller ext4_dirty_ino2
```

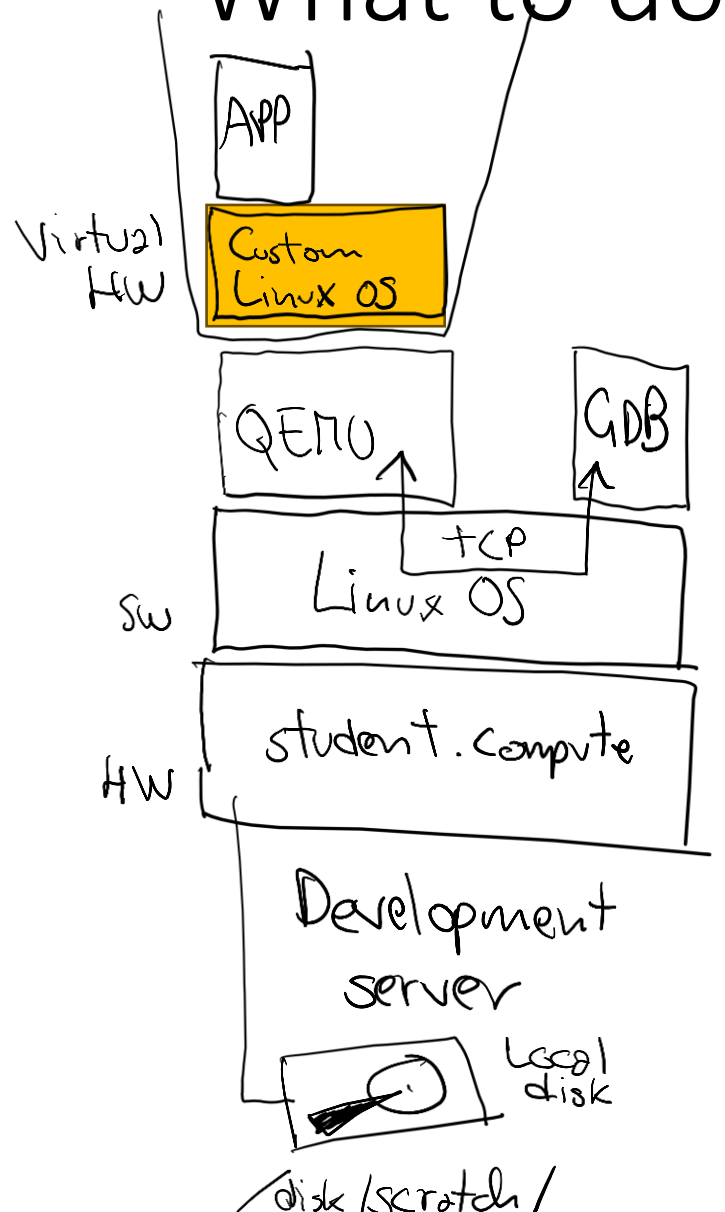
- With -e, trace-cmd records the **events** related to ext4 subsystem

trace-cmd **second example**

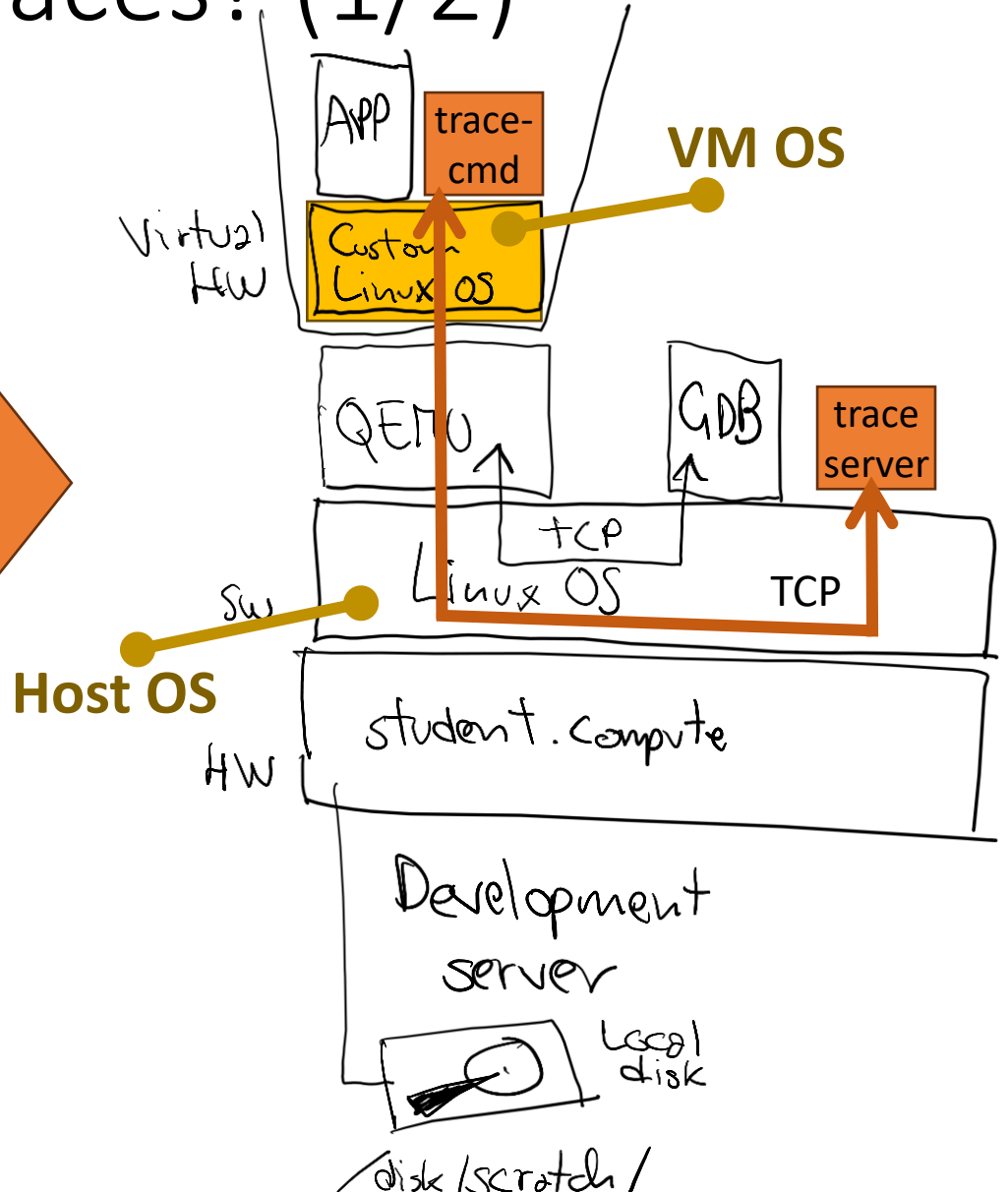
- Let's trace **scheduler's events**
  - `# trace-cmd record -e sched_switch –e sched_wakeup –e irq sleep 10`
  - `# trace-cmd report`

```
unattended-upgr-222   [000]    80.497144: sched_switch:         unattended-upgr:222 [120] R ==> dbus-da]
    dbus-daemon-212   [000]    80.497790: sched_wakeup:         systemd-logind:214 [120] CPU:000
    dbus-daemon-212   [000]    80.497874: sched_switch:         dbus-daemon:212 [120] S ==> systemd-log]
         <idle>-0     [001]    80.498040: softirq_raise:        vec=7 [action=SCHED]
         <idle>-0     [001]    80.498042: softirq_entry:        vec=7 [action=SCHED]
         <idle>-0     [001]    80.498101: softirq_exit:         vec=7 [action=SCHED]
         <idle>-0     [002]    80.498123: sched_switch:         swapper/2:0 [120] R ==> unattended-upgr]
unattended-upgr-222   [002]    80.498194: sched_switch:         unattended-upgr:222 [120] S ==> swapper]
 systemd-logind-214   [000]    80.498292: sched_wakeup:         dbus-daemon:212 [120] CPU:000
 systemd-logind-214   [000]    80.498360: sched_switch:         systemd-logind:214 [120] S ==> dbus-dae]
    dbus-daemon-212   [000]    80.498831: sched_wakeup:         unattended-upgr:222 [120] CPU:000
    dbus-daemon-212   [000]    80.498897: sched_switch:         dbus-daemon:212 [120] S ==> unattended-]
unattended-upgr-222   [000]    80.499535: sched_switch:         unattended-upgr:222 [120] S ==> swapper]
         <idle>-0     [001]    80.696159: softirq_raise:        vec=1 [action=TIMER]
         <idle>-0     [001]    80.696169: softirq_raise:        vec=7 [action=SCHED]
         <idle>-0     [001]    80.696176: softirq_entry:        vec=1 [action=TIMER]
         <idle>-0     [001]    80.696187: sched_wakeup:         kcompactd0:38 [120] CPU:001
         <idle>-0     [001]    80.696190: softirq_exit:         vec=1 [action=TIMER]
         <idle>-0     [001]    80.696191: softirq_entry:        vec=7 [action=SCHED]
         <idle>-0     [001]    80.696201: softirq_exit:         vec=7 [action=SCHED]
         <idle>-0     [001]    80.696216: sched_switch:         swapper/1:0 [120] R ==> kcompactd0:38 []
     kcompactd0-38    [001]    80.696230: sched_switch:         kcompactd0:38 [120] S ==> swapper/1:0 []
         <idle>-0     [001]    81.200466: softirq_raise:        vec=1 [action=TIMER]
         <idle>-0     [001]    81.200472: softirq_raise:        vec=7 [action=SCHED]
         <idle>-0     [001]    81.200478: softirq_entry:        vec=1 [action=TIMER]
         <idle>-0     [001]    81.200487: sched_wakeup:         kcompactd0:38 [120] CPU:001
```

What to do for **very long** traces? (1/2)



Add **trace server**



Install trace server (on host) (1/2)

- We don’t have the **privileges** to install on the host
  - We need to compile a set of libraries and install them **out of tree**
    - `libtraceevent`
    - `libtracefs`
    - `trace-cmd`

- We provide a script
  - Execute it into a new directory
    - `sXXXXXXX: cd /disk/scratch/sXXXXXXX/`
    - `sXXXXXXX: mkdir tracing`
    - `sXXXXXXX: cd tracing`
    - `sXXXXXXX: wget https://pastebin.com/raw/b3YJ9ZRU`
    - `sXXXXXXX: mv b3YJ9ZRU tscript.sh`
    - `sXXXXXXX: chmod +x tscript.sh`
    - `sXXXXXXX: ./tscript.sh`

Instal

- We do
  - We
    -
    -
    -
- We pr
  - Exec
    -
    -
    -

```
#!/bin/bash

#this script should execute on HOST

ROOT_DIR=root
BASE_DIR=`pwd`
echo "Install path: $BASE_DIR/$ROOT_DIR"

#create directory structure
mkdir $ROOT_DIR

#compile and install libtraceevents
git clone git://git.kernel.org/pub/scm/libs/libtrace/libtraceevent.git
cd libtraceevent
make
make DESTDIR=/$BASE_DIR/$ROOT_DIR install
cd $BASE_DIR

#update env variables
export PKG_CONFIG_PATH=$PKG_CONFIG_PATH":/$BASE_DIR/$ROOT_DIR/usr/local/lib/x86_64-linux-gnu/pkgconfig/"
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH":/$BASE_DIR/$ROOT_DIR/usr/local/lib64"
export PATH=$PATH":/$BASE_DIR/$ROOT_DIR/usr/local/bin"

#compile and install libtracefs
git clone git://git.kernel.org/pub/scm/libs/libtrace/libtracefs.git
cd libtracefs
LDFLAGS+="-L/$BASE_DIR/$ROOT_DIR/usr/local/lib64" CFLAGS+="-I/$BASE_DIR/$ROOT_DIR/usr/local/include/traceevent
" make
make DESTDIR=/$BASE_DIR/$ROOT_DIR install
cd $BASE_DIR

#compile and install trace-cmd
git clone git://git.kernel.org/pub/scm/utils/trace-cmd/trace-cmd.git
cd trace-cmd
LDFLAGS+="-L/$BASE_DIR/$ROOT_DIR/usr/local/lib64" CFLAGS+="-I/$BASE_DIR/$ROOT_DIR/usr/local/include/traceevent
 -I/$BASE_DIR/$ROOT_DIR/usr/local/include/tracefs" make
LDFLAGS+="-L/$BASE_DIR/$ROOT_DIR/usr/local/lib64" CFLAGS+="-I/$BASE_DIR/$ROOT_DIR/usr/local/include/traceevent
 -I/$BASE_DIR/$ROOT_DIR/usr/local/include/tracefs" make libs

make DESTDIR=/$BASE_DIR/$ROOT_DIR install
make DESTDIR=/$BASE_DIR/$ROOT_DIR install_libs
cd $BASE_DIR

echo "Install end, please check for errors, and execute the followings:"
echo "export PKG_CONFIG_PATH=\$PKG_CONFIG_PATH:/$BASE_DIR/$ROOT_DIR/usr/local/lib/x86_64-linux-gnu/pkgconfig:"
echo "export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/$BASE_DIR/$ROOT_DIR/usr/local/lib64:"
echo "export PATH=\$PATH:/$BASE_DIR/$ROOT_DIR/usr/local/bin:"
```

Test the trace server (1/2)

- On the **host**
  - `sXXXXXXX: mkdir images`
  - `sXXXXXXX: trace-cmd listen -p 12345 -d images`
- On the **VM**
  - `# dhclient`
  - `# trace-cmd record -N YYYY.inf.ed.ac.uk:12345 --compression none -e sched_switch -e sched_wakeup -e irq sleep 10`
    - Where YYYY is the name of the host machine
- On the **host**
  - `sXXXXXXX: cd images`
  - `sXXXXXXX: trace-cmd report ABCDFGH.dat`
    - Where ABCDFGH is the file name just produced

Test the

- On the **hos**
  - sXXXXX
  - sXXXXX
- On the **VM**
  - # dhcl
  - # trac
    none -
    - Whe
- On the **hos**
  - sXXXXX
  - sXXXXX
    - Whe

```
     kworker/2:0-300   [002] d..2.  3909.352480: sched_switch:         prev_comm=kworker/2:0 prev_pid=300 prev
_prio=120 prev_state=I ==> next_comm=swapper/2 next_pid=0 next_prio=120
          <idle>-0     [003] d.h1.  3909.352489: irq_handler_entry:    irq=15 name=ata_piix
          <idle>-0     [003] d.h1.  3909.352531: irq_handler_exit:     irq=15 ret=handled
          <idle>-0     [003] d.h1.  3909.352543: irq_handler_entry:    irq=15 name=ata_piix
          <idle>-0     [003] d.h1.  3909.352561: irq_handler_exit:     irq=15 ret=handled
          <idle>-0     [002] d.h1.  3909.352572: softirq_raise:        vec=4 [action=BLOCK]
          <idle>-0     [002] ..s1.  3909.352575: softirq_entry:        vec=4 [action=BLOCK]
          <idle>-0     [002] dNs5.  3909.352599: sched_wakeup:         comm=kworker/2:1 pid=57 prio=120 target
_cpu=002
          <idle>-0     [002] .Ns1.  3909.352603: softirq_exit:         vec=4 [action=BLOCK]
          <idle>-0     [002] d..2.  3909.352618: sched_switch:         prev_comm=swapper/2 prev_pid=0 prev_pri
o=120 prev_state=R ==> next_comm=kworker/2:1 next_pid=57 next_prio=120
     kworker/2:1-57    [002] d..2.  3909.352643: sched_switch:         prev_comm=kworker/2:1 prev_pid=57 prev_
prio=120 prev_state=I ==> next_comm=swapper/2 next_pid=0 next_prio=120
          <idle>-0     [003] d.h1.  3909.352961: softirq_raise:        vec=1 [action=TIMER]
          <idle>-0     [003] d.h1.  3909.352966: softirq_raise:        vec=7 [action=SCHED]
          <idle>-0     [003] ..s1.  3909.352973: softirq_entry:        vec=1 [action=TIMER]
          <idle>-0     [003] ..s1.  3909.352979: softirq_exit:         vec=1 [action=TIMER]
          <idle>-0     [003] ..s1.  3909.352980: softirq_entry:        vec=7 [action=SCHED]
          <idle>-0     [003] ..s1.  3909.352988: softirq_exit:         vec=7 [action=SCHED]
          <idle>-0     [002] dNh4.  3909.425365: sched_wakeup:         comm=sleep pid=315 prio=120 target_cpu=
002
          <idle>-0     [002] d..2.  3909.425385: sched_switch:         prev_comm=swapper/2 prev_pid=0 prev_pri
o=120 prev_state=R ==> next_comm=sleep next_pid=315 next_prio=120
           <...>-315   [002] d.h..  3909.426013: softirq_raise:        vec=7 [action=SCHED]
           <...>-315   [002] ..s..  3909.426056: softirq_entry:        vec=7 [action=SCHED]
          <idle>-0     [000] d.h1.  3909.426064: softirq_raise:        vec=7 [action=SCHED]
          <idle>-0     [000] ..s1.  3909.426067: softirq_entry:        vec=7 [action=SCHED]
           <...>-315   [002] ..s..  3909.426071: softirq_exit:         vec=7 [action=SCHED]
          <idle>-0     [000] ..s1.  3909.426074: softirq_exit:         vec=7 [action=SCHED]
           <...>-315   [002] d..2.  3909.427833: sched_switch:         prev_comm=sleep prev_pid=315 prev_prio=
120 prev_state=Z ==> next_comm=swapper/2 next_pid=0 next_prio=120
          <idle>-0     [001] dNh2.  3909.427848: sched_wakeup:         comm=trace-cmd pid=310 prio=120 target_
cpu=001
          <idle>-0     [001] d..2.  3909.427880: sched_switch:         prev_comm=swapper/1 prev_pid=0 prev_pri
o=120 prev_state=R ==> next_comm=trace-cmd next_pid=310 next_prio=120
          <idle>-0     [002] d.h1.  3909.427983: softirq_raise:        vec=9 [action=RCU]
          <idle>-0     [002] d.h1.  3909.427992: softirq_raise:        vec=7 [action=SCHED]
          <idle>-0     [002] ..s1.  3909.428010: softirq_entry:        vec=7 [action=SCHED]
          <idle>-0     [002] ..s1.  3909.428033: softirq_exit:         vec=7 [action=SCHED]
          <idle>-0     [002] ..s1.  3909.428034: softirq_entry:        vec=9 [action=RCU]
          <idle>-0     [002] ..s1.  3909.428060: softirq_exit:         vec=9 [action=RCU]
           <...>-310   [001] d.h..  3909.428069: softirq_raise:        vec=7 [action=SCHED]
          <idle>-0     [003] dNh2.  3909.428083: sched_wakeup:         comm=rcu_preempt pid=13 prio=120 target
_cpu=003
           <...>-310   [001] ..s..  3909.428105: softirq_entry:        vec=7 [action=SCHED]
          <idle>-0     [003] d..2.  3909.428126: sched_switch:         prev_comm=swapper/3 prev_pid=0 prev_pri
o=120 prev_state=R ==> next_comm=rcu_preempt next_pid=13 next_prio=120
     rcu_preempt-13    [003] d..2.  3909.428153: sched_switch:         prev_comm=rcu_preempt prev_pid=13 prev_
prio=120 prev_state=I ==> next_comm=swapper/3 next_pid=0 next_prio=120
           <...>-310   [001] ..s..  3909.428157: softirq_exit:         vec=7 [action=SCHED]
[hendry]abarbala:
```

ression