# See what your computer is doing

## With Ftrace utilities

Steven Rostedt
Open Source Engineer
rostedt@goodmis.org / srostedt@vmware.com

Twitter: @VMWopensource / Expo floor #31

## This is a tutorial

Please follow along to get the most out of it

Most distributions have Ftrace configured in their kernels

- I’m using the default Debian “testing” kernel

  (4.18.0-3-amd64)

Must be “root”

- No “sudo” for now

Helpful utilities for later:

- trace-cmd
- KernelShark

```
git clone /media/<user>/FTRACETUT/trace-cmd.git
```

## Where do we find ftrace?

Most distributions mount: /sys/kernel/debug

- find the “tracing” directory there

Can also mount the tracing directory directly

- mount -t tracefs nodev /sys/kernel/tracing

We’ll use /sys/kernel/tracing here

## Let’s look at the tracing directory

```
# mount -t tracefs nodev /sys/kernel/tracing
# cd /sys/kernel/tracing
# ls

available_events            per_cpu              stack_trace
available_filter_functions  printk_formats       stack_trace_filter
available_tracers           README               timestamp_mode
buffer_size_kb              saved_cmdlines       trace
buffer_total_size_kb        saved_cmdlines_size  trace_clock
current_tracer              saved_tgids          trace_marker
dyn_ftrace_total_info       set_event            trace_marker_raw
enabled_functions           set_event_pid        trace_options
events                      set_ftrace_filter    trace_pipe
free_buffer                 set_ftrace_notrace   tracing_cpumask
instances                   set_ftrace_pid       tracing_max_latency
kprobe_events               set_graph_function   tracing_on
kprobe_profile              set_graph_notrace    tracing_thresh
max_graph_depth             snapshot             uprobe_events
options                     stack_max_size       uprobe_profile
```

## README

Actually lets you know if some things are available

- Not everything, but features are only in here if they exist in the kernel

Has basic reference on how to perform some features

Has basic information to some of the files in tracefs

## available_tracers

A “tracer” performs some action

- nop - No action (no tracer enabled)
- function - Trace kernel functions
- function_graph - Graph kernel functions
- blk - Used with blktrace (not discussed in this tutorial)
- mmiotrace - Trace interactions between drivers and hardware (not discussed in this tutorial)

## Other available_tracers

These tracers are not available in my Debian kernel

- hwlat - Detects hardware latency (i.e. SMI interrupts)
- wakeup{,_dl,_rt} - Records max wake up latency
- irqsoff - Records max interrupt latency
- preemptoff - Records max preemption disabled latency
- preemptirqsoff - Records max preemption or interrupt disabled latency

We wont talk about these in this tutorial either

- But they are good to know about

## current_tracer

The way to enable a tracer

```
# echo function > current_tracer
# echo nop > current_tracer
```

## trace

Where to see the output of the trace

```
# echo function > current_tracer
# cat trace

# tracer: function
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
     soffice.bin-1850  [003] .... 1277817.313622: __fget_light <-sockfd_lookup_light
     soffice.bin-1850  [003] .... 1277817.313622: __fget <-__fget_light
     soffice.bin-1850  [003] .... 1277817.313623: ___sys_recvmsg <-__sys_recvmsg
     soffice.bin-1850  [003] .... 1277817.313623: copy_msghdr_from_user <-___sys_recvmsg
     soffice.bin-1850  [003] .... 1277817.313624: rw_copy_check_uvector <-import_iovec
     soffice.bin-1850  [003] .... 1277817.313624: __check_object_size <-rw_copy_check_uvector
     soffice.bin-1850  [003] .... 1277817.313624: __virt_addr_valid <-__check_object_size
     soffice.bin-1850  [003] .... 1277817.313625: check_stack_object <-__check_object_size
     soffice.bin-1850  [003] .... 1277817.313625: sock_recvmsg <-___sys_recvmsg
     soffice.bin-1850  [003] .... 1277817.313625: security_socket_recvmsg <-sock_recvmsg
     soffice.bin-1850  [003] .... 1277817.313626: apparmor_socket_recvmsg <-security_socket_recvmsg
```

## trace

Remember to turn off the tracer

```
# echo nop > current_tracer
# cat trace

# tracer: nop
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
```

## tracing_on

Stop updates to the trace without clearing it.

Note, tracing is still happening, just not recording (will have overhead)

```
# echo function > current_tracer
# echo 0 > tracing_on
# cat trace

# tracer: function
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
           xfwm4-17105 [001] d... 1278457.346868: hrtick_update <-ttwu_do_activate
           xfwm4-17105 [001] d... 1278457.346868: ttwu_do_wakeup <-try_to_wake_up
           xfwm4-17105 [001] d... 1278457.346869: check_preempt_curr <-ttwu_do_wakeup
           xfwm4-17105 [001] d... 1278457.346869: resched_curr <-check_preempt_curr
           xfwm4-17105 [001] d... 1278457.346870: ttwu_stat <-try_to_wake_up
           xfwm4-17105 [001] d... 1278457.346870: _raw_spin_unlock_irqrestore <-try_to_wake_up
           xfwm4-17105 [001] d... 1278457.346871: _raw_spin_unlock_irqrestore <-ep_poll_callback
           xfwm4-17105 [001] d... 1278457.346871: _raw_spin_unlock_irqrestore <-__wake_up_common_lock
```

## tracing_on  (Caution)

Writing ASCII “0” into the file, stops tracing

Writing ASCII “1” into the file starts tracing

Note, be sure to add a space between the number and ‘>’

Do this:

```
# echo 0 > tracing_on
# echo 1 > tracing_on
```

Not this:

```
# echo 0>tracing_on
# echo 1>tracing_on
```

## tracing_on  (Caution)

Writing ASCII “0” into the file, stops tracing

Writing ASCII “1” into the file starts tracing

Note, be sure to add a space between the number and ‘>’

Do this:

```
# echo 0 > tracing_on
# echo 1 > tracing_on
```

Not this:

```
# echo 0>tracing_on
# echo 1>tracing_on
```

**Why?**

## tracing_on  (Caution)

```
# echo 0>tracing_on
# echo 1>tracing_on
```

Writes standard input and standard output into tracing_on respectively

## trace_marker

Writes into the ftrace ring buffer (along with other kernel events)

```
# echo nop > current_tracer
# echo 1 > tracing_on
# echo ‘hello world!’ > trace_marker
# cat trace

# tracer: nop
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
            bash-13441 [003] .... 1279231.118528: tracing_mark_write: hello world!
```

## trace_marker

In C code:

```
static int marker_fd = -1;

static void setup_trace_marker(void)
{
     marker_fd = open("/sys/kernel/tracing/trace_marker", O_WRONLY);
}

static void write_trace_marker(const char *fmt, ...)
{
     char buf[BUFSIZ];
     va_list ap;
     int n;

     if (marker_fd < 0)
          return;

     va_start(ap, fmt);
     n = vsnprintf(buf, BUFSIZ, fmt, ap);
     va_end(ap);

     write(marker_fd, buf, n);
}
```

## Function tracing

Trace almost any function in the kernel!

- See the possible functions in “available_filter_functions”

Limit what functions you trace

- set_ftrace_filter - Only trace these functions
- set_ftrace_notrace - Do not trace these functions

  (**notrace** takes precedence over **filter**)

## set_ftrace_filter and set_ftrace_notrace

Just echo function names into the files

```
# echo foo > set_ftrace_notrace
```

Can add more than one at a time (white space delimited)

```
# echo foo bar > set_ftrace_filter
```

Append with the bash concatenation “>>”

```
# echo zoot >> set_ftrace_filter
```

Clear with just writing nothing into it

```
# echo > set_ftrace_notrace
```

## set_ftrace_filter and set_ftrace_notrace

Can handle minor wild cards “*” and “?”

```
# echo ‘?lock*d’ > set_ftrace_filter
```

Can use available_filter_functions for more complex filtering

```
# cut -d’ ‘ -f1 available_filter_functions | grep -E ‘ipv(4|6)’ > set_ftrace_filter
```

- Note, the ‘cut’ is to remove module names:

```
nf_reject_ip_tcphdr_get [nf_reject_ipv4]
```

# Function tracing

```
# echo '*lock*' > set_ftrace_filter
# echo '*clock*' > set_ftrace_notrace
# echo function > current_tracer
# cat trace

# tracer: function
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
            Xorg-16967 [002] .... 1383242.709737: mutex_trylock <-i915_vma_retire
            Xorg-16967 [002] .... 1383242.709737: ww_mutex_unlock <-i915_vma_retire
            Xorg-16967 [002] .... 1383242.709737: mutex_unlock <-i915_vma_retire
            Xorg-16967 [002] .... 1383242.709738: _raw_spin_lock <-i915_vma_retire
            Xorg-16967 [002] .... 1383242.709738: mutex_trylock <-i915_vma_retire
            Xorg-16967 [002] .... 1383242.709738: ww_mutex_unlock <-i915_vma_retire
            Xorg-16967 [002] .... 1383242.709739: mutex_unlock <-i915_vma_retire
            Xorg-16967 [002] .... 1383242.709739: _raw_spin_lock <-i915_vma_retire
            Xorg-16967 [002] .... 1383242.709740: mutex_trylock <-i915_vma_retire
            Xorg-16967 [002] .... 1383242.709740: ww_mutex_unlock <-i915_vma_retire
            Xorg-16967 [002] .... 1383242.709740: mutex_unlock <-i915_vma_retire
            Xorg-16967 [002] .... 1383242.709741: _raw_spin_lock <-i915_vma_retire
            Xorg-16967 [002] .... 1383242.709741: mutex_trylock <-i915_vma_retire
            Xorg-16967 [002] .... 1383242.709741: ww_mutex_unlock <-i915_vma_retire
```

## Function Graph tracing

Similar to function tracing (traces the same functions)

Also traces the return of a function

Allows to see a graph of the function calls

- What function called which function

# Function Graph tracing

```
# echo function_graph > current_tracer
# cat trace

# tracer: function_graph
#
# CPU  DURATION                  FUNCTION CALLS
# |     |   |                     |   |   |   |
 2)   0.032 us    |        } /* fput */
 2)   7.984 us    |      } /* __sys_recvmsg */
 2)   8.259 us    |    } /* __x64_sys_recvmsg */
 2)   8.558 us    |  } /* do_syscall_64 */
 2)               |  do_syscall_64() {
 2)               |    __x64_sys_poll() {
 2)               |      poll_select_set_timeout() {
 2)   0.104 us    |        ktime_get_ts64();
 2)   0.045 us    |        timespec64_add_safe();
 2)   0.738 us    |      }
 2)               |      do_sys_poll() {
 2)               |        __check_object_size() {
 2)   0.034 us    |          __virt_addr_valid();
 2)   0.036 us    |          check_stack_object();
 2)   0.613 us    |        }
 2)               |        select_estimate_accuracy() {
 2)   0.053 us    |          ktime_get_ts64();
 2)   0.033 us    |          set_normalized_timespec64();
 2)   0.681 us    |        }
 2)               |        __fdget() {
 2)               |          __fget_light() {
 2)   0.069 us    |            __fget();
```

# Function Graph tracing

```
# echo function_graph > current_tracer
# cat trace

# tracer: function_graph
#
# CPU  DURATION                  FUNCTION CALLS
# |     |   |                     |   |   |   |
 2)   0.032 us    |        } /* fput */
 2)   7.984 us    |      } /* __sys_recvmsg */
 2)   8.259 us    |    } /* __x64_sys_recvmsg */
 2)   8.558 us    |  } /* do_syscall_64 */
 2)               |  do_syscall_64() {
 2)               |    __x64_sys_poll() {
 2)               |      poll_select_set_timeout() {
 2)   0.104 us    |        ktime_get_ts64();
 2)   0.045 us    |        timespec64_add_safe();
 2)   0.738 us    |      }
 2)               |      do_sys_poll() {
 2)               |        __check_object_size() {
 2)   0.034 us    |          __virt_addr_valid();
 2)   0.036 us    |          check_stack_object();
 2)   0.613 us    |        }
 2)               |        select_estimate_accuracy() {
 2)   0.053 us    |          ktime_get_ts64();
 2)   0.033 us    |          set_normalized_timespec64();
 2)   0.681 us    |        }
 2)               |        __fdget() {
 2)               |          __fget_light() {
 2)   0.069 us    |            __fget();
```

## Function Graph tracing

Don’t trust all the times (just take it as a guide)

Function graph tracer adds overhead

- You see the overhead of functions traced within other functions

Closest to actual time is a single entity

- The enter and exit of a function are together
- Denoted by singe event with ‘;’
- Instead of two events with ‘{‘ and ‘}’

The set_ftrace_filter and set_ftrace_notrace affect function_graph

## Function Graph tracing

```
# echo do_IRQ > set_ftrace_filter
# echo function_graph > current_tracer
# cat trace

# tracer: function_graph
#
# CPU  DURATION                  FUNCTION CALLS
# |     |   |                     |   |   |   |
 2)   ==========> |
 2) + 14.098 us   |  do_IRQ();
 2)   <========== |
 2)   ==========> |
 2)   6.920 us    |  do_IRQ();
 2)   <========== |
 2)   ==========> |
 2)   6.259 us    |  do_IRQ();
 2)   <========== |
 2)   ==========> |
 2)   5.625 us    |  do_IRQ();
 2)   <========== |
 2)   ==========> |
 2) + 10.433 us   |  do_IRQ();
 2)   <========== |
 2)   ==========> |
 2) + 11.347 us   |  do_IRQ();
 2)   <========== |
 2)   ==========> |
 2) + 11.230 us   |  do_IRQ();
```

## Tracing Events

Function tracing is great BUT!

- It doesn’t show much data besides the functions being called
- No parameters or variables can be seen

Which brings us to Trace Events

- They are points in the kernel that write into the trace buffer
- Record specific data within  the kernel
- Allows to see more detailed view of what is happening

## Tracing Events

They are grouped by “system”

- sched - schedule events
- irq - interrupt events
- net - networking events
- syscalls - system call events (all system calls and their parameters)
- module - module loading, unloading, freeing, etc
- kvm - events for the KVM guests
- exceptions - page faults
- cgroup - changes to cgroups
- And many many more!

## Trace Event Systems

```
# ls events

alarmtimer  fib6            iommu        napi            rseq       udp
asoc        filelock        irq          net             rtc        v4l2
block       filemap         irq_matrix   nmi             sched      vb2
bridge      fs_dax          irq_vectors  oom             scsi       vmscan
btrfs       ftrace          jbd2         page_isolation  signal     vsyscall
cfg80211    gpio            kmem         pagemap         skb        wbt
cgroup      hda             kvm          percpu          smbus      workqueue
clk         hda_controller  kvmmmu       power           sock       writeback
compaction  hda_intel       libata       printk          spi        x86_fpu
cpuhp       header_event    mac80211     qdisc           sunrpc     xdp
devlink     header_page     mce          random          swiotlb    xen
dma_fence   huge_memory     mei          ras             syscalls   xfs
drm         hyperv          migrate      raw_syscalls    task       xhci-hcd
enable      i2c             mmc          rcu             tcp
exceptions  i915            module       regmap          thermal
ext4        initcall        mpx          regulator       timer
fib         intel-sst       msr          rpm             tlb
```

# Tracing Events

Each system has several events

For example: The “sched” system

- sched_switch
- sched_waking
- sched_process_fork
- sched_process_exit
- sched_process_exec
- sched_migrate_task
- etc

## Trace Events

```
# ls events/sched

enable                      sched_process_fork  sched_stick_numa
filter                      sched_process_free  sched_swap_numa
sched_kthread_stop          sched_process_hang  sched_switch
sched_kthread_stop_ret      sched_process_wait  sched_wait_task
sched_migrate_task          sched_stat_blocked  sched_wake_idle_without_ipi
sched_move_numa             sched_stat_iowait   sched_wakeup
sched_pi_setprio            sched_stat_runtime  sched_wakeup_new
sched_process_exec          sched_stat_sleep    sched_waking
sched_process_exit          sched_stat_wait
```

# Trace Events

```
# ls events/sched

enable                    sched_process_fork  sched_stick_numa
filter                    sched_process_free  sched_swap_numa
sched_kthread_stop        sched_process_hang  sched_switch
sched_kthread_stop_ret    sched_process_wait  sched_wait_task
sched_migrate_task        sched_stat_blocked  sched_wake_idle_without_ipi
sched_move_numa           sched_stat_iowait   sched_wakeup
sched_pi_setprio          sched_stat_runtime  sched_wakeup_new
sched_process_exec        sched_stat_sleep    sched_waking
sched_process_exit        sched_stat_wait
```

# Enabling Trace Events

```
# echo nop > current_tracer
# echo 1 > events/sched/sched_waking/enable
# echo 1 > events/sched/sched_switch/enable
# cat trace

# tracer: nop
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
           Timer-19729 [000] d.h. 1555860.848766: sched_waking: comm=ModuleProcessTh pid=24498 prio=120 target_cpu=000
           Timer-19729 [000] d... 1555860.848777: sched_waking: comm=Web Content pid=19688 prio=120 target_cpu=002
           Timer-19729 [000] d... 1555860.848818: sched_switch: prev_comm=Timer prev_pid=19729 prev_prio=120 prev_state=D ==> next_comm=swapper/0 next_pid=0 next_prio=120
          <idle>-0     [000] d... 1555860.848862: sched_switch: prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=S ==> next_comm=Xorg next_pid=16967 next_prio=120
            Xorg-16967 [000] d... 1555860.848917: sched_switch: prev_comm=Xorg prev_pid=16967 prev_prio=120 prev_state=D ==> next_comm=swapper/0 next_pid=0 next_prio=120
          <idle>-0     [000] d.h. 1555860.850480: sched_waking: comm=SoftwareVsyncTh pid=19636 prio=120 target_cpu=000
          <idle>-0     [000] d... 1555860.850513: sched_switch: prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=S ==> next_comm=SoftwareVsyncTh next_pid=19636 next_prio
 SoftwareVsyncTh-19636 [000] d... 1555860.850562: sched_waking: comm=IPDL Background pid=19629 prio=120 target_cpu=002
 SoftwareVsyncTh-19636 [000] d... 1555860.850604: sched_switch: prev_comm=SoftwareVsyncTh prev_pid=19636 prev_prio=120 prev_state=D ==> next_comm=swapper/0 next_pid=0 next_prio
          <idle>-0     [000] d... 1555860.850629: sched_switch: prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=S ==> next_comm=Gecko_IOThread next_pid=19602 next_prio=
  Gecko_IOThread-19602 [000] d... 1555860.850677: sched_waking: comm=Chrome_ChildThr pid=19690 prio=120 target_cpu=001
  Gecko_IOThread-19602 [000] d... 1555860.850701: sched_switch: prev_comm=Gecko_IOThread prev_pid=19602 prev_prio=120 prev_state=D ==> next_comm=swapper/0 next_pid=0 next_prio=
          <idle>-0     [000] d.h. 1555860.861078: sched_waking: comm=InputThread pid=16979 prio=120 target_cpu=001
          <idle>-0     [000] d... 1555860.861311: sched_switch: prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=S ==> next_comm=Xorg next_pid=16967 next_prio=120
            Xorg-16967 [000] d... 1555860.861384: sched_switch: prev_comm=Xorg prev_pid=16967 prev_prio=120 prev_state=D ==> next_comm=swapper/0 next_pid=0 next_prio=120
          <idle>-0     [000] d.h. 1555860.863389: sched_waking: comm=SendControllerT pid=5426 prio=120 target_cpu=000
          <idle>-0     [000] d... 1555860.863418: sched_switch: prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=S ==> next_comm=SendControllerT next_pid=5426 next_prio=
 SendControllerT-5426  [000] d... 1555860.863480: sched_switch: prev_comm=SendControllerT prev_pid=5426 prev_prio=120 prev_state=D ==> next_comm=swapper/0 next_pid=0 next_prio=
          <idle>-0     [000] d.h. 1555860.863524: sched_waking: comm=SendControllerT pid=5426 prio=120 target_cpu=000
          <idle>-0     [000] d... 1555860.863535: sched_switch: prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=S ==> next_comm=SendControllerT next_pid=5426 next_prio=
 SendControllerT-5426  [000] d... 1555860.863571: sched_waking: comm=TaskSchedulerFo pid=27478 prio=120 target_cpu=001
```

# Enabling Trace Events

```
sched_waking: comm=ModuleProcessTh pid=24498 prio=120 target_cpu=000
sched_waking: comm=Web Content pid=19688 prio=120 target_cpu=002
sched_switch: prev_comm=Timer prev_pid=19729 prev_prio=120 prev_state=D ==> next_comm=swapper/0 next_pid=0 next_prio=120
sched_switch: prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=S ==> next_comm=Xorg next_pid=16967 next_prio=120
sched_switch: prev_comm=Xorg prev_pid=16967 prev_prio=120 prev_state=D ==> next_comm=swapper/0 next_pid=0 next_prio=120
sched_waking: comm=SoftwareVsyncTh pid=19636 prio=120 target_cpu=000
sched_switch: prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=S ==> next_comm=SoftwareVsyncTh next_pid=19636 next_p
sched_waking: comm=IPDL Background pid=19629 prio=120 target_cpu=002
sched_switch: prev_comm=SoftwareVsyncTh prev_pid=19636 prev_prio=120 prev_state=D ==> next_comm=swapper/0 next_pid=0 next_p
sched_switch: prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=S ==> next_comm=Gecko_IOThread next_pid=19602 next_pr
sched_waking: comm=Chrome_ChildThr pid=19690 prio=120 target_cpu=001
sched_switch: prev_comm=Gecko_IOThread prev_pid=19602 prev_prio=120 prev_state=D ==> next_comm=swapper/0 next_pid=0 next_pr
sched_waking: comm=InputThread pid=16979 prio=120 target_cpu=001
sched_switch: prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=S ==> next_comm=Xorg next_pid=16967 next_prio=120
sched_switch: prev_comm=Xorg prev_pid=16967 prev_prio=120 prev_state=D ==> next_comm=swapper/0 next_pid=0 next_prio=120
sched_waking: comm=SendControllerT pid=5426 prio=120 target_cpu=000
sched_switch: prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=S ==> next_comm=SendControllerT next_pid=5426 next_pr
sched_switch: prev_comm=SendControllerT prev_pid=5426 prev_prio=120 prev_state=D ==> next_comm=swapper/0 next_pid=0 next_pr
sched_waking: comm=SendControllerT pid=5426 prio=120 target_cpu=000
sched_switch: prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=S ==> next_comm=SendControllerT next_pid=5426 next_pr
sched_waking: comm=TaskSchedulerFo pid=27478 prio=120 target_cpu=001
```

## Enabling Groups of Events

```
# echo nop > current_tracer
# echo 1 > events/sched/enable
# cat trace

# tracer: nop
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
    firefox.real-19591 [002] d... 1558025.460824: sched_stat_runtime: comm=JS Helper pid=19608 runtime=7550 [ns] vruntime=111608395027855 [ns]
    firefox.real-19591 [002] d... 1558025.460826: sched_wakeup: comm=JS Helper pid=19607 prio=120 target_cpu=000
    firefox.real-19591 [002] d... 1558025.460827: sched_waking: comm=JS Helper pid=19606 prio=120 target_cpu=000
    firefox.real-19591 [002] d... 1558025.460829: sched_stat_runtime: comm=JS Helper pid=19607 runtime=2756 [ns] vruntime=111608395161002 [ns]
    firefox.real-19591 [002] d... 1558025.460830: sched_wakeup: comm=JS Helper pid=19606 prio=120 target_cpu=000
    firefox.real-19591 [002] d... 1558025.460831: sched_waking: comm=JS Helper pid=19611 prio=120 target_cpu=000
    firefox.real-19591 [002] d... 1558025.460833: sched_stat_runtime: comm=JS Helper pid=19606 runtime=2174 [ns] vruntime=111608394990889 [ns]
    firefox.real-19591 [002] d... 1558025.460834: sched_wakeup: comm=JS Helper pid=19611 prio=120 target_cpu=000
    firefox.real-19591 [002] d... 1558025.460835: sched_waking: comm=JS Helper pid=19609 prio=120 target_cpu=000
    firefox.real-19591 [002] d... 1558025.460837: sched_stat_runtime: comm=JS Helper pid=19611 runtime=2218 [ns] vruntime=111608394992422 [ns]
    firefox.real-19591 [002] d... 1558025.460838: sched_wakeup: comm=JS Helper pid=19609 prio=120 target_cpu=000
    firefox.real-19591 [002] d... 1558025.460839: sched_waking: comm=JS Helper pid=19608 prio=120 target_cpu=000
    firefox.real-19591 [002] d... 1558025.460841: sched_stat_runtime: comm=JS Helper pid=19609 runtime=2484 [ns] vruntime=111608394997215 [ns]
    firefox.real-19591 [002] d... 1558025.460841: sched_wakeup: comm=JS Helper pid=19608 prio=120 target_cpu=000
    firefox.real-19591 [002] d... 1558025.461027: sched_waking: comm=JS Helper pid=19608 prio=120 target_cpu=000
    firefox.real-19591 [002] d... 1558025.461029: sched_wake_idle_without_ipi: cpu=0
    firefox.real-19591 [002] d... 1558025.461030: sched_wakeup: comm=JS Helper pid=19608 prio=120 target_cpu=000
    firefox.real-19591 [002] d... 1558025.461030: sched_waking: comm=JS Helper pid=19606 prio=120 target_cpu=001
    firefox.real-19591 [002] d... 1558025.461032: sched_wake_idle_without_ipi: cpu=1
    firefox.real-19591 [002] d... 1558025.461032: sched_wakeup: comm=JS Helper pid=19606 prio=120 target_cpu=001
    firefox.real-19591 [002] d... 1558025.461032: sched_waking: comm=JS Helper pid=19611 prio=120 target_cpu=000
```

# Enabling All Events!

```
# echo 1 > events/enable
# cat trace

# tracer: nop
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
      hostd-fair-2334  [003] d... 1558119.421939: tlb_flush: pages:0 reason:flush on task switch (0)
      hostd-fair-2334  [003] d... 1558119.421940: x86_fpu_regs_deactivated: x86/fpu: 00000000047327aa initialized: 1 xfeatures: 3 xcomp_bv: 800000000000001f
          <idle>-0     [003] d... 1558119.421942: cpu_idle: state=2 cpu_id=3
          <idle>-0     [003] dN.. 1558119.422000: cpu_idle: state=4294967295 cpu_id=3
          <idle>-0     [003] dN.. 1558119.422001: rcu_utilization: Start context switch
          <idle>-0     [003] dN.. 1558119.422001: rcu_utilization: End context switch
          <idle>-0     [003] d... 1558119.422003: sched_switch: prev_comm=swapper/3 prev_pid=0 prev_prio=120 prev_state=S ==> next_comm=hostd-fair next_pid=
          <idle>-0     [003] d... 1558119.422003: tlb_flush: pages:0 reason:flush on task switch (0)
          <idle>-0     [003] d... 1558119.422004: write_msr: c0000100, value 7f8741977700
          <idle>-0     [003] d... 1558119.422004: x86_fpu_regs_activated: x86/fpu: 00000000047327aa initialized: 1 xfeatures: 3 xcomp_bv: 800000000000001f
      hostd-fair-2334  [003] .... 1558119.422005: sys_exit: NR 202 = 0
      hostd-fair-2334  [003] .... 1558119.422005: sys_futex -> 0x0
      hostd-fair-2334  [003] .... 1558119.422007: sys_enter: NR 202 (5558d741eae0, 81, 1, 0, 0, 5558d6f7bcd8)
      hostd-fair-2334  [003] .... 1558119.422007: sys_futex(uaddr: 5558d741eae0, op: 81, val: 1, utime: 0, uaddr2: 0, val3: 5558d6f7bcd8)
      hostd-fair-2334  [003] .... 1558119.422008: sys_exit: NR 202 = 0
      hostd-fair-2334  [003] .... 1558119.422008: sys_futex -> 0x0
      hostd-fair-2334  [003] .... 1558119.422010: sys_enter: NR 202 (5558d741eb3c, 80, 0, 0, 0, 5558d6f7bcd8)
      hostd-fair-2334  [003] .... 1558119.422011: sys_futex(uaddr: 5558d741eb3c, op: 80, val: 0, utime: 0, uaddr2: 0, val3: 5558d6f7bcd8)
      hostd-fair-2334  [003] d... 1558119.422011: rcu_utilization: Start context switch
      hostd-fair-2334  [003] d... 1558119.422012: rcu_utilization: End context switch
      hostd-fair-2334  [003] d... 1558119.422012: sched_stat_runtime: comm=hostd-fair pid=2334 runtime=13739 [ns] vruntime=80967984047 [ns]
```

# What’s in an event?

```
# cat events/sched/sched_switch/format

name: sched_switch
ID: 312
format:
      field:unsigned short common_type;   offset:0;   size:2;     signed:0;
      field:unsigned char common_flags;   offset:2;   size:1;     signed:0;
      field:unsigned char common_preempt_count; offset:3;   size:1;     signed:0;
      field:int common_pid;   offset:4;   size:4;     signed:1;

      field:char prev_comm[16];     offset:8;   size:16;    signed:1;
      field:pid_t prev_pid;   offset:24;  size:4;     signed:1;
      field:int prev_prio;    offset:28;  size:4;     signed:1;
      field:long prev_state;  offset:32;  size:8;     signed:1;
      field:char next_comm[16];     offset:40;  size:16;    signed:1;
      field:pid_t next_pid;   offset:56;  size:4;     signed:1;
      field:int next_prio;    offset:60;  size:4;     signed:1;

print fmt: "prev_comm=%s prev_pid=%d prev_prio=%d prev_state=%s%s ==> next_comm=%s next_pid=%d
next_prio=%d", REC->prev_comm, REC->prev_pid, REC->prev_prio, (REC->prev_state & ((((0x0000 | 0x0001 |
0x0002 | 0x0004 | 0x0008 | 0x0010 | 0x0020 | 0x0040) + 1) << 1) - 1)) ? __print_flags(REC->prev_state
& ((((0x0000 | 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010 | 0x0020 | 0x0040) + 1) << 1) - 1), "|",
{ 0x01, "S" }, { 0x02, "D" }, { 0x04, "T" }, { 0x08, "t" }, { 0x10, "X" }, { 0x20, "Z" }, { 0x40,
"P" }, { 0x80, "I" }) : "R", REC->prev_state & (((0x0000 | 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010
| 0x0020 | 0x0040) + 1) << 1) ? "+" : "", REC->next_comm, REC->next_pid, REC->next_prio
```

## What’s in an event?

```
# cat events/sched/sched_switch/format

name: sched_switch
ID: 312
format:
      field:unsigned short common_type;   offset:0;   size:2;     signed:0;
      field:unsigned char common_flags;   offset:2;   size:1;     signed:0;
      field:unsigned char common_preempt_count; offset:3;   size:1;     signed:0;
      field:int common_pid;   offset:4;   size:4;     signed:1;

      field:char prev_comm[16];     offset:8;   size:16;    signed:1;
      field:pid_t prev_pid;   offset:24;  size:4;     signed:1;
      field:int prev_prio;    offset:28;  size:4;     signed:1;
      field:long prev_state;  offset:32;  size:8;     signed:1;
      field:char next_comm[16];     offset:40;  size:16;    signed:1;
      field:pid_t next_pid;   offset:56;  size:4;     signed:1;
      field:int next_prio;    offset:60;  size:4;     signed:1;

print fmt: "prev_comm=%s prev_pid=%d prev_prio=%d prev_state=%s%s ==> next_comm=%s next_pid=%d
next_prio=%d", REC->prev_comm, REC->prev_pid, REC->prev_prio, (REC->prev_state & ((((0x0000 | 0x0001 |
0x0002 | 0x0004 | 0x0008 | 0x0010 | 0x0020 | 0x0040) + 1) << 1) - 1)) ? __print_flags(REC->prev_state
& ((((0x0000 | 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010 | 0x0020 | 0x0040) + 1) << 1) - 1), "|",
{ 0x01, "S" }, { 0x02, "D" }, { 0x04, "T" }, { 0x08, "t" }, { 0x10, "X" }, { 0x20, "Z" }, { 0x40,
"P" }, { 0x80, "I" }) : "R", REC->prev_state & (((0x0000 | 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010
| 0x0020 | 0x0040) + 1) << 1) ? "+" : "", REC->next_comm, REC->next_pid, REC->next_prio
```

# What’s in an event?

```
# cat events/sched/sched_switch/format

name: sched_switch
ID: 312
format:
	field:unsigned short common_type;	offset:0;	size:2;	signed:0;
	field:unsigned char common_flags;	offset:2;	size:1;	signed:0;
	field:unsigned char common_preempt_count;	offset:3;	size:1;	signed:0;
	field:int common_pid;	offset:4;	size:4;	signed:1;

	field:char prev_comm[16];	offset:8;	size:16;	signed:1;
	field:pid_t prev_pid;	offset:24;	size:4;	signed:1;
	field:int prev_prio;	offset:28;	size:4;	signed:1;
	field:long prev_state;	offset:32;	size:8;	signed:1;
	field:char next_comm[16];	offset:40;	size:16;	signed:1;
	field:pid_t next_pid;	offset:56;	size:4;	signed:1;
	field:int next_prio;	offset:60;	size:4;	signed:1;

print fmt: "prev_comm=%s prev_pid=%d prev_prio=%d prev_state=%s%s ==> next_comm=%s next_pid=%d
next_prio=%d", REC->prev_comm, REC->prev_pid, REC->prev_prio, (REC->prev_state & ((((0x0000 | 0x0001 |
0x0002 | 0x0004 | 0x0008 | 0x0010 | 0x0020 | 0x0040) + 1) << 1) - 1)) ? __print_flags(REC->prev_state
& ((((0x0000 | 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010 | 0x0020 | 0x0040) + 1) << 1) - 1), "|",
{ 0x01, "S" }, { 0x02, "D" }, { 0x04, "T" }, { 0x08, "t" }, { 0x10, "X" }, { 0x20, "Z" }, { 0x40,
"P" }, { 0x80, "I" }) : "R", REC->prev_state & (((0x0000 | 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010
| 0x0020 | 0x0040) + 1) << 1) ? "+" : "", REC->next_comm, REC->next_pid, REC->next_prio
```

## What’s in an event?

```
# cat events/sched/sched_switch/format

name: sched_switch
ID: 312
format:
	field:unsigned short common_type;	offset:0;	size:2;	signed:0;
	field:unsigned char common_flags;	offset:2;	size:1;	signed:0;
	field:unsigned char common_preempt_count;	offset:3;	size:1;	signed:0;
	field:int common_pid;	offset:4;	size:4;	signed:1;

	field:char prev_comm[16];	offset:8;	size:16;	signed:1;
	field:pid_t prev_pid;	offset:24;	size:4;	signed:1;
	field:int prev_prio;	offset:28;	size:4;	signed:1;
	field:long prev_state;	offset:32;	size:8;	signed:1;
	field:char next_comm[16];	offset:40;	size:16;	signed:1;
	field:pid_t next_pid;	offset:56;	size:4;	signed:1;
	field:int next_prio;	offset:60;	size:4;	signed:1;

print fmt: "prev_comm=%s prev_pid=%d prev_prio=%d prev_state=%s%s ==> next_comm=%s next_pid=%d
next_prio=%d", REC->prev_comm, REC->prev_pid, REC->prev_prio, (REC->prev_state & ((((0x0000 | 0x0001 |
0x0002 | 0x0004 | 0x0008 | 0x0010 | 0x0020 | 0x0040) + 1) << 1) - 1)) ? __print_flags(REC->prev_state
& ((((0x0000 | 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010 | 0x0020 | 0x0040) + 1) << 1) - 1), "|",
{ 0x01, "S" }, { 0x02, "D" }, { 0x04, "T" }, { 0x08, "t" }, { 0x10, "X" }, { 0x20, "Z" }, { 0x40,
"P" }, { 0x80, "I" }) : "R", REC->prev_state & (((0x0000 | 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010
| 0x0020 | 0x0040) + 1) << 1) ? "+" : "", REC->next_comm, REC->next_pid, REC->next_prio
```

## What’s in an event?

```
print fmt: "prev_comm=%s prev_pid=%d prev_prio=%d prev_state=%s%s ==> next_comm=%s next_pid=%d
next_prio=%d",
            REC->prev_comm, REC->prev_pid, REC->prev_prio,

            (REC->prev_state & ((((0x0000 | 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010 | 0x0020 |
0x0040) + 1) << 1) - 1)) ?
             __print_flags(REC->prev_state & ((((0x0000 | 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010 |
0x0020 | 0x0040) + 1) << 1) - 1), "|",
                       { 0x01, "S" },
                       { 0x02, "D" },
                       { 0x04, "T" },
                       { 0x08, "t" },
                       { 0x10, "X" },
                       { 0x20, "Z" },
                       { 0x40, "P" },
                       { 0x80, "I" }) :
             "R",

            REC->prev_state & (((0x0000 | 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010 | 0x0020 |
0x0040) + 1) << 1) ? "+" : "",
            REC->next_comm, REC->next_pid, REC->next_prio
```

# What’s in an event?

```
print fmt: "prev_comm=%s prev_pid=%d prev_prio=%d prev_state=%s%s ==> next_comm=%s next_pid=%d
next_prio=%d",
            REC->prev_comm, REC->prev_pid, REC->prev_prio,

            (REC->prev_state & (TASK_REPORT_MAX - 1)) ?
              __print_flags(REC->prev_state & (TASK_REPORT_MAX - 1), "|",
                        { TASK_INTERRUPTIBLE, "S" },
                        { TASK_UNINTERRUPTIBLE, "D" },
                        { __TASK_STOPPED, "T" },
                        { __TASK_TRACED, "t" },
                        { EXIT_DEAD, "X" },
                        { EXIT_ZOMBIE, "Z" },
                        { TASK_PARKED, "P" },
                        { TASK_DEAD, "I" }) :
              "R",

            REC->prev_state & TASK_REPORT_MAX ? "+" : "",
            REC->next_comm, REC->next_pid, REC->next_prio
```

## Filtering Events

Too Much Info!

- This can be just as bad as not enough info
- Finding the needle in the haystack
- Signal to noise!

Filter out everything we do not want

- Just trace what we are interested in

The “filter” file

## What’s in an event?

```
# cat events/sched/sched_switch/format

name: sched_switch
ID: 312
format:
      field:unsigned short common_type;   offset:0;   size:2;     signed:0;
      field:unsigned char common_flags;   offset:2;   size:1;     signed:0;
      field:unsigned char common_preempt_count; offset:3;   size:1;     signed:0;
      field:int common_pid;   offset:4;   size:4;     signed:1;

      field:char prev_comm[16];     offset:8;   size:16;    signed:1;
      field:pid_t prev_pid;   offset:24;  size:4;     signed:1;
      field:int prev_prio;    offset:28;  size:4;     signed:1;
      field:long prev_state;  offset:32;  size:8;     signed:1;
      field:char next_comm[16];     offset:40;  size:16;    signed:1;
      field:pid_t next_pid;   offset:56;  size:4;     signed:1;
      field:int next_prio;    offset:60;  size:4;     signed:1;
```

## Filtering Events

```
# echo 'prev_comm == "bash" && prev_state & 0x02' > events/sched/sched_switch/filter
# echo 1 > events/sched/sched_switch/enable
# echo > trace
# cat trace
```

```
# tracer: nop
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
            bash-27607 [000] d... 1559578.742752: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/0 next_pid=0 next_prio=120
            bash-27607 [001] d... 1559579.026451: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/1 next_pid=0 next_prio=120
            bash-27607 [000] d... 1559579.111236: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/0 next_pid=0 next_prio=120
            bash-27607 [000] d... 1559579.171892: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/0 next_pid=0 next_prio=120
            bash-27607 [002] d... 1559579.304215: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=xfwm4 next_pid=17105 next_prio=120
            bash-27607 [002] d... 1559579.414473: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/2 next_pid=0 next_prio=120
            bash-27607 [002] d... 1559579.470549: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/2 next_pid=0 next_prio=120
            bash-27607 [002] d... 1559579.537746: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/2 next_pid=0 next_prio=120
            bash-27607 [002] d... 1559579.591502: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/2 next_pid=0 next_prio=120
            bash-27607 [003] d... 1559579.695324: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/3 next_pid=0 next_prio=120
            bash-27607 [002] d... 1559579.841453: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/2 next_pid=0 next_prio=120
            bash-27607 [002] d... 1559579.863734: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/2 next_pid=0 next_prio=120
            bash-27607 [003] d... 1559615.733229: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/3 next_pid=0 next_prio=120
            bash-27607 [003] d... 1559615.832504: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/3 next_pid=0 next_prio=120
            bash-27607 [003] d... 1559616.309440: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/3 next_pid=0 next_prio=120
            bash-27607 [003] d... 1559616.778076: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/3 next_pid=0 next_prio=120
```

## Event Triggers

Make a event do something special

- Turn off tracing
- Turn on tracing
- Take a “snapshot”
- Produce a stack dump
- Enable another event
- Disable another event

## Filtered Events along with triggers

```
# echo 'prev_comm == "bash" && prev_state & 0x02' > events/sched/sched_switch/filter
# echo 'stacktrace if prev_comm == "bash" && prev_state & 0x02' > events/sched/sched_switch/trigger
# echo 1 > events/sched/sched_switch/enable
# echo > trace
# cat trace
```

```
# tracer: nop
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
            bash-27607 [002] d... 1559946.989729: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/2 next_pid=0 next_prio=120
            bash-27607 [002] d... 1559946.989746: <stack trace>
 => __schedule
 => schedule
 => schedule_hrtimeout_range_clock
 => poll_schedule_timeout.constprop.11
 => do_select
 => core_sys_select
 => do_pselect
 => __x64_sys_pselect6
 => do_syscall_64
 => entry_SYSCALL_64_after_hwframe
            bash-27607 [001] d... 1559947.145818: sched_switch: prev_comm=bash prev_pid=27607 prev_prio=120 prev_state=D ==> next_comm=swapper/1 next_pid=0 next_prio=120
            bash-27607 [001] d... 1559947.145833: <stack trace>
 => __schedule
 => schedule
 => schedule_hrtimeout_range_clock
 => poll_schedule_timeout.constprop.11
 => do_select
 => core_sys_select
 => do_pselect
```

## Triggers are a little more difficult to remove

```
# echo '!stacktrace' > events/sched/sched_switch/trigger
```

## ftrace - a tool for everyone with BusyBox

Everything we did so far used echo or cat

Makes using ftrace extremely simple on limited systems

But can be very tedious

- Lots of things to remember (what file does what)
- Not very intuitive
- Can be painstaking
- Hard to do batch processing
- Hard to record specific functions

## Introducing trace-cmd

You did make that clone didn’t you?

Is an executable that interacts with the ftrace interface

No need to worry about the tracefs system

- It does it for you

Can also save the data to a file

- Uses per_cpu/cpuX/trace_pipe_raw
- Reads the binary data directly from the ring buffer
- Uses splice(2) to write directly to a file or network (zero copy)

Uses the format files to know how to read the binary data

## Introducing trace-cmd

Forget everything you learned so far

- (no don’t really, but let’s start over)

From now on, be in a directory that you can write to

- Still be root user, or at least start all commands with “sudo”

trace-cmd will do the work for you

If you did “make install_doc”

- man trace-cmd
- man trace-cmd record
- man trace-cmd report
- etc

## trace-cmd start and show

```
# trace-cmd start -e sched_switch -f 'prev_comm == "bash" && prev_state & 0x02’ \
   -R ‘stracktrace if prev_comm == “bash” && prev_state & 0x02’
# trace-cmd show

# tracer: nop
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
            bash-7855  [000] d... 1564606.264976: sched_switch: prev_comm=bash prev_pid=7855 prev_prio=120 prev_state=D ==> next_comm=swapper/0 next_pid=0 next_prio=120
            bash-7855  [000] d... 1564606.264994: <stack trace>
 => __schedule
 => schedule
 => schedule_hrtimeout_range_clock
 => poll_schedule_timeout.constprop.11
 => do_select
 => core_sys_select
 => do_pselect
 => __x64_sys_pselect6
 => do_syscall_64
 => entry_SYSCALL_64_after_hwframe
            bash-7855  [001] d... 1564606.664732: sched_switch: prev_comm=bash prev_pid=7855 prev_prio=120 prev_state=D ==> next_comm=SendControllerT next_pid=12716
next_prio=120
            bash-7855  [001] d... 1564606.664749: <stack trace>
 => __schedule
 => schedule
 => schedule_hrtimeout_range_clock
 => poll_schedule_timeout.constprop.11
 => do_select
 => core_sys_select
 => do_pselect
```

## trace-cmd stat and reset

```
# trace-cmd stat

Events:
 Individual events:
    sched
        sched_switch

Filters:
  sched:sched_switch "(prev_comm == "bash" && prev_state & 0x02)"

Triggers:
  sched:sched_switch "stacktrace:unlimited if prev_comm == "bash" && prev_state & 0x02"

Buffer size in kilobytes (per cpu):
   1408

Buffer total size in kilobytes:
   5632

Tracing is enabled

# trace-cmd reset
# trace-cmd stat

Events:
 All disabled

Buffer size in kilobytes (per cpu):
   1408

Buffer total size in kilobytes:
   5632

Tracing is disabled
```

# trace-cmd start -p nop

```
# trace-cmd reset
# trace-cmd stat

Events:
 All disabled

Buffer size in kilobytes (per cpu):
   1408

Buffer total size in kilobytes:
   5632

Tracing is disabled

# trace-cmd start -p nop
# trace-cmd stat

Events:
 All disabled

Buffer size in kilobytes (per cpu):
   1408

Buffer total size in kilobytes:
   5632

Tracing is enabled
```

# trace-cmd record and start

“start” enables tracing but does no recording

- Use “show” to see the in-kernel ring buffer output “trace” file.

“record” records the trace data into a file (default: trace.dat)

- Use “report” to read the file
- The “record” will zero copy from the per_cpu files

“start” has most the same options as “record”

- To enable tracing
- “-e” for events
- “-p” for tracers (historically they were once called “plugins”)

## Now let’s start seeing what your computer is doing

Isn’t that the title of this tutorial?

We spent enough time on details, let’s start doing something fun

Let’s write a bash script

And see exactly what goes on in the kernel!

# Make a really simple program

Using your favorite editor, create this file

```
#!/bin/bash

echo “Hello world!”
```

## Introduction (or not) to strace

strace - Traces the system calls a program makes

- See how it communicates with the kernel
- It understands the calls that are made
- Shows file names and parameters of the system calls

We will start with this before jumping into the kernel

## strace our Hello World!

```
# strace ./hello &> out
# cat out

execve("./hello", ["./hello"], 0x7ffebab3e880 /* 13 vars */) = 0
brk(NULL)                               = 0x559b65f84000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=156426, ...}) = 0
mmap(NULL, 156426, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f446db12000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libtinfo.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0@\351\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=183528, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f446db10000
[ more mmap calls ]
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libdl.so.2", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0000\21\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=14592, ...}) = 0
[ more mmap calls ]
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\260A\2\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=1824496, ...}) = 0
[ more mmap calls ]
close(3)                                = 0
```

## strace our Hello World!

```
# strace ./hello &> out
# cat out

[..]

openat(AT_FDCWD, "./hello", O_RDONLY)   = 3
stat("./hello", {st_mode=S_IFREG|0755, st_size=33, ...}) = 0
ioctl(3, TCGETS, 0x7ffcfa6a6950)        = -1 ENOTTY (Inappropriate ioctl for device)
lseek(3, 0, SEEK_CUR)                   = 0
read(3, "#!/bin/bash\n\necho 'hello world!'"..., 80) = 33
lseek(3, 0, SEEK_SET)                   = 0
prlimit64(0, RLIMIT_NOFILE, NULL, {rlim_cur=1024, rlim_max=1024*1024}) = 0
fcntl(255, F_GETFD)                     = -1 EBADF (Bad file descriptor)
dup2(3, 255)                            = 255
close(3)                                = 0
fcntl(255, F_SETFD, FD_CLOEXEC)         = 0
fcntl(255, F_GETFL)                     = 0x8000 (flags O_RDONLY|O_LARGEFILE)
fstat(255, {st_mode=S_IFREG|0755, st_size=33, ...}) = 0
lseek(255, 0, SEEK_CUR)                 = 0
read(255, "#!/bin/bash\n\necho 'hello world!'"..., 33) = 33
fstat(1, {st_mode=S_IFCHR|0600, st_rdev=makedev(136, 3), ...}) = 0
write(1, "hello world!\n", 13)          = 13
read(255, "", 33)                       = 0
rt_sigprocmask(SIG_BLOCK, [CHLD], [], 8) = 0
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
exit_group(0)                           = ?
+++ exited with 0 +++
```

## strace our Hello World!

```
# strace ./hello &> out
# cat out

[..]

openat(AT_FDCWD, "./hello", O_RDONLY)   = 3
stat("./hello", {st_mode=S_IFREG|0755, st_size=33, ...}) = 0
ioctl(3, TCGETS, 0x7ffcfa6a6950)        = -1 ENOTTY (Inappropriate ioctl for device)
lseek(3, 0, SEEK_CUR)                   = 0
read(3, "#!/bin/bash\n\necho 'hello world!'"..., 80) = 33
lseek(3, 0, SEEK_SET)                   = 0
prlimit64(0, RLIMIT_NOFILE, NULL, {rlim_cur=1024, rlim_max=1024*1024}) = 0
fcntl(255, F_GETFD)                     = -1 EBADF (Bad file descriptor)
dup2(3, 255)                            = 255
close(3)                                = 0
fcntl(255, F_SETFD, FD_CLOEXEC)         = 0
fcntl(255, F_GETFL)                     = 0x8000 (flags O_RDONLY|O_LARGEFILE)
fstat(255, {st_mode=S_IFREG|0755, st_size=33, ...}) = 0
lseek(255, 0, SEEK_CUR)                 = 0
read(255, "#!/bin/bash\n\necho 'hello world!'"..., 33) = 33
fstat(1, {st_mode=S_IFCHR|0600, st_rdev=makedev(136, 3), ...}) = 0
write(1, "hello world!\n", 13hello world!
)           = 13
read(255, "", 33)                       = 0
rt_sigprocmask(SIG_BLOCK, [CHLD], [], 8) = 0
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
exit_group(0)                           = ?
+++ exited with 0 +++
```

## How do bash scripts work?

```
# strace ./hello &> out
# cat out

[..]

execve("./hello", ["./hello"], 0x7ffebab3e880 /* 13 vars */) = 0

[..]

openat(AT_FDCWD, "./hello", O_RDONLY)   = 3

[..]

read(255, "#!/bin/bash\n\necho 'hello world!'"..., 33) = 33

[..]

write(1, "hello world!\n", 13)          = 13
```

## Let’s have a look at execve

In the kernel, system calls start with arch specific headers

__x64_sys_execve “__x64_sys_” is added to the system call name

```
# trace-cmd list -f execve

audit_log_execve_info
__do_execve_file.isra.35
__ia32_compat_sys_execve
__ia32_compat_sys_execveat
__ia32_sys_execve
__ia32_sys_execveat
__x32_compat_sys_execve
__x64_sys_execve
__x64_sys_execveat
__x32_compat_sys_execveat
do_execve_file
do_execve
do_execveat
```

## trace-cmd list -f execve?

For more information, do: “man trace-cmd list”

## trace-cmd list -f execve?

For more information, do: “man trace-cmd list”

Moving on, we don’t have time for details!

## trace-cmd list -f execve?

For more information, do: “man trace-cmd list”

Moving on, we don’t have time for details!

All you need to know, is that we found our execve system call function

## Graph our Hello World!

After all that, I cheat with “*sys_exec*”!

```
# trace-cmd record -p function_graph -g ‘*sys_exec*’ ./hello

  plugin 'function_graph'
hello world!
CPU0 data recorded at offset=0x652000
    0 bytes in size
CPU1 data recorded at offset=0x652000
    851968 bytes in size
CPU2 data recorded at offset=0x722000
    0 bytes in size
CPU3 data recorded at offset=0x722000
    12288 bytes in size
```

# Graph our Hello World!

```
# trace-cmd report

CPU 0 is empty
CPU 2 is empty
cpus=4
           hello-16873 [003] 1653533.226805: funcgraph_entry:                   |  __x64_sys_execve() {
           hello-16873 [003] 1653533.226808: funcgraph_entry:                   |    getname() {
           hello-16873 [003] 1653533.226808: funcgraph_entry:                   |      getname_flags() {
           hello-16873 [003] 1653533.226809: funcgraph_entry:                   |        kmem_cache_alloc() {
           hello-16873 [003] 1653533.226809: funcgraph_entry:                   |          _cond_resched() {
           hello-16873 [003] 1653533.226809: funcgraph_entry:        0.054 us   |            rcu_all_qs();
           hello-16873 [003] 1653533.226809: funcgraph_exit:         0.431 us   |          }
           hello-16873 [003] 1653533.226809: funcgraph_entry:        0.032 us   |          should_failslab();
           hello-16873 [003] 1653533.226810: funcgraph_entry:        0.038 us   |          prefetch_freepointer();
           hello-16873 [003] 1653533.226810: funcgraph_entry:        0.045 us   |          memcg_kmem_put_cache();
           hello-16873 [003] 1653533.226810: funcgraph_exit:         1.611 us   |        }
           hello-16873 [003] 1653533.226810: funcgraph_entry:                   |        __check_object_size() {
           hello-16873 [003] 1653533.226811: funcgraph_entry:        0.062 us   |          __virt_addr_valid();
           hello-16873 [003] 1653533.226811: funcgraph_entry:        0.043 us   |          __check_heap_object();
           hello-16873 [003] 1653533.226811: funcgraph_entry:        0.042 us   |          check_stack_object();
           hello-16873 [003] 1653533.226812: funcgraph_exit:         1.023 us   |        }
           hello-16873 [003] 1653533.226812: funcgraph_exit:         3.253 us   |      }
           hello-16873 [003] 1653533.226812: funcgraph_exit:         3.583 us   |    }
           hello-16873 [003] 1653533.226812: funcgraph_entry:                   |    __do_execve_file.isra.35() {
           hello-16873 [003] 1653533.226812: funcgraph_entry:                   |      unshare_files() {
           hello-16873 [003] 1653533.226812: funcgraph_entry:        0.048 us   |        unshare_fd();
           hello-16873 [003] 1653533.226813: funcgraph_exit:         0.354 us   |      }
           hello-16873 [003] 1653533.226813: funcgraph_entry:                   |      kmem_cache_alloc_trace() {
           hello-16873 [003] 1653533.226813: funcgraph_entry:                   |        _cond_resched() {
           hello-16873 [003] 1653533.226813: funcgraph_entry:        0.032 us   |          rcu_all_qs();
           hello-16873 [003] 1653533.226813: funcgraph_exit:         0.310 us   |        }
           hello-16873 [003] 1653533.226814: funcgraph_entry:        0.030 us   |        should_failslab();
           hello-16873 [003] 1653533.226814: funcgraph_entry:        0.106 us   |        prefetch_freepointer();
           hello-16873 [003] 1653533.226814: funcgraph_entry:        0.119 us   |        memcg_kmem_put_cache();
           hello-16873 [003] 1653533.226815: funcgraph_exit:         1.845 us   |      }
```

## There’s a lot of info here!

```
# trace-cmd report | wc -l
16870
```

## There's a lot of info here!

```
# trace-cmd report | wc -l
16870
```

Interrupts do happen

```
# trace-cmd report -I | wc -l
10196
```

# There’s a lot of info here!

```
# trace-cmd report | wc -l
16870
```

Interrupts do happen

```
# trace-cmd report -I | wc -l
10196
```

As well as “soft” interrupts

```
# trace-cmd report -IS | wc -l
10092
```

## Only trace a few functions down

```
# trace-cmd record -p function_graph -g ‘*sys_exec*’ --max-graph-depth 5 ./hello

  plugin 'function_graph'
hello world!
CPU0 data recorded at offset=0x652000
    4096 bytes in size
CPU1 data recorded at offset=0x653000
    0 bytes in size
CPU2 data recorded at offset=0x653000
    0 bytes in size
CPU3 data recorded at offset=0x653000
    16384 bytes in size
```

## More likely prevents interrupts from being recorded

```
# trace-cmd report | wc -l
347
# trace-cmd report -I | wc -l
347
# trace-cmd report -IS | wc -l
347
```

## Remove “_cond_resched” calls (too many)

```
# trace-cmd record -p function_graph -g ‘*sys_exec*’ --max-graph-depth 5 \
  -n _cond_resched ./hello

  plugin 'function_graph'
hello world!
CPU0 data recorded at offset=0x652000
    4096 bytes in size
CPU1 data recorded at offset=0x653000
    0 bytes in size
CPU2 data recorded at offset=0x653000
    0 bytes in size
CPU3 data recorded at offset=0x653000
    12288 bytes in size
```

## Much better

```
# trace-cmd report | wc -l
244
# trace-cmd report -I | wc -l
242
# trace-cmd report -IS | wc -l
242
```

## And let’s just do 4 deep

```
# trace-cmd record -p function_graph -g ‘*sys_exec*’ --max-graph-depth 4 \
  -n _cond_resched ./hello

  plugin 'function_graph'
hello world!
CPU0 data recorded at offset=0x652000
    0 bytes in size
CPU1 data recorded at offset=0x652000
    4096 bytes in size
CPU2 data recorded at offset=0x653000
    0 bytes in size
CPU3 data recorded at offset=0x653000
    8192 bytes in size
```

# Easier to understand

```
# trace-cmd report -IS

CPU 0 is empty
CPU 2 is empty
cpus=4
           hello-18623 [001] 1654799.238555: funcgraph_entry:                   |  __x64_sys_execve() {
           hello-18623 [001] 1654799.238561: funcgraph_entry:                   |    getname() {
           hello-18623 [001] 1654799.238561: funcgraph_entry:                   |      getname_flags() {
           hello-18623 [001] 1654799.238562: funcgraph_entry:        0.373 us   |        kmem_cache_alloc();
           hello-18623 [001] 1654799.238563: funcgraph_entry:        0.232 us   |        __check_object_size();
           hello-18623 [001] 1654799.238563: funcgraph_exit:         1.613 us   |      }
           hello-18623 [001] 1654799.238563: funcgraph_exit:         2.040 us   |    }
           hello-18623 [001] 1654799.238564: funcgraph_entry:                   |    __do_execve_file.isra.35() {
           hello-18623 [001] 1654799.238564: funcgraph_entry:                   |    smp_irq_work_interrupt() {
           hello-18623 [001] 1654799.238565: funcgraph_entry:                   |      irq_enter() {
           hello-18623 [001] 1654799.238565: funcgraph_entry:        0.094 us   |        rcu_irq_enter();
           hello-18623 [001] 1654799.238572: funcgraph_entry:        0.053 us   |        idle_cpu();
           hello-18623 [001] 1654799.238573: funcgraph_entry:        0.109 us   |        rcu_irq_exit();
           hello-18623 [001] 1654799.238573: funcgraph_exit:         1.058 us   |      }
           hello-18623 [001] 1654799.238574: funcgraph_exit:         8.978 us   |    }
           hello-18623 [001] 1654799.238574: funcgraph_entry:                   |      unshare_files() {
           hello-18623 [001] 1654799.238575: funcgraph_entry:        0.079 us   |        unshare_fd();
           hello-18623 [001] 1654799.238575: funcgraph_exit:         0.625 us   |      }
           hello-18623 [001] 1654799.238575: funcgraph_entry:                   |      kmem_cache_alloc_trace() {
           hello-18623 [001] 1654799.238576: funcgraph_entry:        0.057 us   |        should_failslab();
           hello-18623 [001] 1654799.238577: funcgraph_entry:        0.065 us   |        prefetch_freepointer();
           hello-18623 [001] 1654799.238577: funcgraph_entry:        0.064 us   |        memcg_kmem_put_cache();
           hello-18623 [001] 1654799.238578: funcgraph_exit:         2.141 us   |      }
           hello-18623 [001] 1654799.238578: funcgraph_entry:                   |      prepare_bprm_creds() {
           hello-18623 [001] 1654799.238578: funcgraph_entry:        0.247 us   |        mutex_lock_interruptible();
           hello-18623 [001] 1654799.238579: funcgraph_entry:        1.410 us   |        prepare_exec_creds();
           hello-18623 [001] 1654799.238580: funcgraph_exit:         2.375 us   |      }
           hello-18623 [001] 1654799.238581: funcgraph_entry:        0.049 us   |      _raw_spin_lock();
```

# Easier to understand

```
# trace-cmd report -IS

CPU 0 is empty
CPU 2 is empty
cpus=4
           hello-18623 [001] 1654799.238555: funcgraph_entry:                   |  __x64_sys_execve() {
           hello-18623 [001] 1654799.238561: funcgraph_entry:                   |    getname() {
           hello-18623 [001] 1654799.238561: funcgraph_entry:                   |      getname_flags() {
           hello-18623 [001] 1654799.238562: funcgraph_entry:        0.373 us   |        kmem_cache_alloc();
           hello-18623 [001] 1654799.238563: funcgraph_entry:        0.232 us   |        __check_object_size();
           hello-18623 [001] 1654799.238563: funcgraph_exit:         1.613 us   |      }
           hello-18623 [001] 1654799.238563: funcgraph_exit:         2.040 us   |    }
           hello-18623 [001] 1654799.238564: funcgraph_entry:                   |    __do_execve_file.isra.35() {
           hello-18623 [001] 1654799.238564: funcgraph_entry:                   |    smp_irq_work_interrupt() {
           hello-18623 [001] 1654799.238565: funcgraph_entry:                   |      irq_enter() {
           hello-18623 [001] 1654799.238565: funcgraph_entry:        0.094 us   |        rcu_irq_enter();
           hello-18623 [001] 1654799.238572: funcgraph_entry:        0.053 us   |        idle_cpu();
           hello-18623 [001] 1654799.238573: funcgraph_entry:        0.109 us   |        rcu_irq_exit();
           hello-18623 [001] 1654799.238573: funcgraph_exit:         1.058 us   |      }
           hello-18623 [001] 1654799.238574: funcgraph_exit:         8.978 us   |    }
           hello-18623 [001] 1654799.238574: funcgraph_entry:                   |      unshare_files() {
           hello-18623 [001] 1654799.238575: funcgraph_entry:        0.079 us   |        unshare_fd();
           hello-18623 [001] 1654799.238575: funcgraph_exit:         0.625 us   |      }
           hello-18623 [001] 1654799.238575: funcgraph_entry:                   |      kmem_cache_alloc_trace() {
           hello-18623 [001] 1654799.238576: funcgraph_entry:        0.057 us   |        should_failslab();
           hello-18623 [001] 1654799.238577: funcgraph_entry:        0.065 us   |        prefetch_freepointer();
           hello-18623 [001] 1654799.238577: funcgraph_entry:        0.064 us   |        memcg_kmem_put_cache();
           hello-18623 [001] 1654799.238578: funcgraph_exit:         2.141 us   |      }
           hello-18623 [001] 1654799.238578: funcgraph_entry:                   |      prepare_bprm_creds() {
           hello-18623 [001] 1654799.238578: funcgraph_entry:        0.247 us   |        mutex_lock_interruptible();
           hello-18623 [001] 1654799.238579: funcgraph_entry:        1.410 us   |        prepare_exec_creds();
           hello-18623 [001] 1654799.238580: funcgraph_exit:         2.375 us   |      }
           hello-18623 [001] 1654799.238581: funcgraph_entry:        0.049 us   |      _raw_spin_lock();
```

# Easier to understand

```
   hello-18623   1.... 1654799.238581: funcgraph_entry:                   |      do_open_execat() {
   hello-18623   1.... 1654799.238581: funcgraph_entry:        3.846 us   |        do_filp_open();
   hello-18623   1.... 1654799.238585: funcgraph_entry:        0.035 us   |        __fsnotify_parent();
   hello-18623   1.... 1654799.238586: funcgraph_entry:        0.041 us   |        fsnotify();
   hello-18623   1.... 1654799.238586: funcgraph_exit:         4.887 us   |      }
   hello-18623   1.... 1654799.238586: funcgraph_entry:                   |      sched_exec() {
   hello-18623   1.... 1654799.238586: funcgraph_entry:        0.036 us   |        _raw_spin_lock_irqsave();
   hello-18623   1d... 1654799.238587: funcgraph_entry:        1.016 us   |        select_task_rq_fair();
   hello-18623   1d... 1654799.238588: funcgraph_entry:        0.044 us   |        _raw_spin_unlock_irqrestore();
   hello-18623   1.... 1654799.238588: funcgraph_entry:      + 11.455 us  |        stop_one_cpu();
   hello-18623   3d... 1654799.238602: funcgraph_entry:        1.061 us   |        smp_irq_work_interrupt();
   hello-18623   3.... 1654799.238603: funcgraph_exit:       + 17.049 us  |      }
   hello-18623   3.... 1654799.238604: funcgraph_entry:                   |      mm_alloc() {
   hello-18623   3.... 1654799.238604: funcgraph_entry:        0.809 us   |        kmem_cache_alloc();
   hello-18623   3.... 1654799.238605: funcgraph_entry:        2.296 us   |        mm_init();
   hello-18623   3.... 1654799.238608: funcgraph_exit:         3.806 us   |      }
   hello-18623   3.... 1654799.238608: funcgraph_entry:        0.041 us   |      _raw_spin_lock();
   hello-18623   3.... 1654799.238608: funcgraph_entry:                   |      vm_area_alloc() {
   hello-18623   3.... 1654799.238608: funcgraph_entry:        0.706 us   |        kmem_cache_alloc();
   hello-18623   3.... 1654799.238609: funcgraph_exit:         1.001 us   |      }
   hello-18623   3.... 1654799.238609: funcgraph_entry:        0.222 us   |      down_write_killable();
   hello-18623   3.... 1654799.238610: funcgraph_entry:        0.045 us   |      vm_get_page_prot();
   hello-18623   3.... 1654799.238610: funcgraph_entry:                   |      insert_vm_struct() {
   hello-18623   3.... 1654799.238610: funcgraph_entry:        0.310 us   |        security_vm_enough_memory_mm();
   hello-18623   3.... 1654799.238611: funcgraph_entry:        0.347 us   |        vma_link();
   hello-18623   3.... 1654799.238612: funcgraph_exit:         1.295 us   |      }
   hello-18623   3.... 1654799.238612: funcgraph_entry:        0.037 us   |      up_write();
   hello-18623   3.... 1654799.238612: funcgraph_entry:        0.382 us   |      count.isra.24.constprop.38();
   hello-18623   3.... 1654799.238613: funcgraph_entry:        1.817 us   |      count.isra.24.constprop.38();
   hello-18623   3.... 1654799.238615: funcgraph_entry:                   |      prepare_binprm() {
   hello-18623   3.... 1654799.238615: funcgraph_entry:        0.216 us   |        mnt_may_suid();
   hello-18623   3.... 1654799.238615: funcgraph_entry:        9.257 us   |        security_bprm_set_creds();
   hello-18623   3.... 1654799.238625: funcgraph_entry:        2.590 us   |        kernel_read();
   hello-18623   3.... 1654799.238628: funcgraph_exit:       + 12.984 us  |      }
```

## Easier to understand

```
   hello-18623   3.... 1654799.238628: funcggraph_entry:                  |      copy_strings_kernel() {
   hello-18623   3.... 1654799.238628: funcgraph_entry:      + 10.840 us   |        copy_strings.isra.26();
   hello-18623   3.... 1654799.238639: funcgraph_exit:       + 11.232 us   |      }
   hello-18623   3.... 1654799.238640: funcgraph_entry:                    |      copy_strings.isra.26() {
   hello-18623   3.... 1654799.238640: funcgraph_entry:        0.549 us    |        get_user_pages_remote();
   hello-18623   3.... 1654799.238641: funcgraph_entry:        0.090 us    |        __check_object_size();
[..]
   hello-18623   3.... 1654799.238648: funcgraph_entry:        0.085 us    |        __check_object_size();
   hello-18623   3.... 1654799.238649: funcgraph_exit:         8.993 us    |      }
   hello-18623   3.... 1654799.238649: funcgraph_entry:                    |      copy_strings.isra.26() {
   hello-18623   3.... 1654799.238649: funcgraph_entry:        0.449 us    |        get_user_pages_remote();
   hello-18623   3.... 1654799.238650: funcgraph_entry:        0.089 us    |        __check_object_size();
   hello-18623   3.... 1654799.238650: funcgraph_exit:         1.444 us    |      }
   hello-18623   3.... 1654799.238650: funcgraph_entry:                    |      would_dump() {
   hello-18623   3.... 1654799.238651: funcgraph_entry:        0.119 us    |        inode_permission();
   hello-18623   3.... 1654799.238651: funcgraph_exit:         0.415 us    |      }
   hello-18623   3.... 1654799.238651: funcgraph_entry:        0.114 us    |      task_active_pid_ns();
   hello-18623   3.... 1654799.238652: funcgraph_entry:        0.079 us    |      __task_pid_nr_ns();
   hello-18623   3.... 1654799.238652: funcgraph_entry:                    |      search_binary_handler() {
   hello-18623   3.... 1654799.238652: funcgraph_entry:        0.048 us    |        security_bprm_check();
   hello-18623   3.... 1654799.238652: funcgraph_entry:        0.057 us    |        _raw_read_lock();
   hello-18623   3.... 1654799.238653: funcgraph_entry:        0.036 us    |        try_module_get();
   hello-18623   3.... 1654799.238653: funcgraph_entry:      ! 214.953 us  |        load_script();
   hello-18623   3.... 1654799.238869: funcgraph_entry:        0.052 us    |        _raw_read_lock();
   hello-18623   3.... 1654799.238869: funcgraph_entry:        0.034 us    |        module_put();
   hello-18623   3.... 1654799.238869: funcgraph_exit:       ! 217.327 us  |      }
   hello-18623   3.... 1654799.238870: funcgraph_entry:        0.066 us    |      proc_exec_connector();
   hello-18623   3.... 1654799.238870: funcgraph_entry:                    |      acct_update_integrals() {
   hello-18623   3d... 1654799.238870: funcgraph_entry:        0.042 us    |        __acct_update_integrals();
   hello-18623   3.... 1654799.238870: funcgraph_exit:         0.336 us    |      }
   hello-18623   3.... 1654799.238871: funcgraph_entry:                    |      task_numa_free() {
   hello-18623   3.... 1654799.238871: funcgraph_entry:        0.040 us    |        kfree();
   hello-18623   3d... 1654799.238872: funcgraph_entry:        3.233 us    |        smp_irq_work_interrupt();
   hello-18623   3.... 1654799.238876: funcgraph_exit:         5.134 us    |      }
```

# Easier to understand

```
   hello-18623   3.... 1654799.238628: funcgraph_entry:                   |      copy_strings_kernel() {
   hello-18623   3.... 1654799.238628: funcgraph_entry:      + 10.840 us   |        copy_strings.isra.26();
   hello-18623   3.... 1654799.238639: funcgraph_exit:       + 11.232 us   |      }
   hello-18623   3.... 1654799.238640: funcgraph_entry:                   |      copy_strings.isra.26() {
   hello-18623   3.... 1654799.238640: funcgraph_entry:        0.549 us   |        get_user_pages_remote();
   hello-18623   3.... 1654799.238641: funcgraph_entry:        0.090 us   |        __check_object_size();
[..]
   hello-18623   3.... 1654799.238648: funcgraph_entry:        0.085 us   |        __check_object_size();
   hello-18623   3.... 1654799.238649: funcgraph_exit:         8.993 us   |      }
   hello-18623   3.... 1654799.238649: funcgraph_entry:                   |      copy_strings.isra.26() {
   hello-18623   3.... 1654799.238649: funcgraph_entry:        0.449 us   |        get_user_pages_remote();
   hello-18623   3.... 1654799.238650: funcgraph_entry:        0.089 us   |        __check_object_size();
   hello-18623   3.... 1654799.238650: funcgraph_exit:         1.444 us   |      }
   hello-18623   3.... 1654799.238650: funcgraph_entry:                   |      would_dump() {
   hello-18623   3.... 1654799.238651: funcgraph_entry:        0.119 us   |        inode_permission();
   hello-18623   3.... 1654799.238651: funcgraph_exit:         0.415 us   |      }
   hello-18623   3.... 1654799.238651: funcgraph_entry:        0.114 us   |      task_active_pid_ns();
   hello-18623   3.... 1654799.238652: funcgraph_entry:        0.079 us   |      __task_pid_nr_ns();
   hello-18623   3.... 1654799.238652: funcgraph_entry:                   |      search_binary_handler() {
   hello-18623   3.... 1654799.238652: funcgraph_entry:        0.048 us   |        security_bprm_check();
   hello-18623   3.... 1654799.238652: funcgraph_entry:        0.057 us   |        _raw_read_lock();
   hello-18623   3.... 1654799.238653: funcgraph_entry:        0.036 us   |        try_module_get();
   hello-18623   3.... 1654799.238653: funcgraph_entry:      ! 214.953 us |        load_script();
   hello-18623   3.... 1654799.238869: funcgraph_entry:        0.052 us   |        _raw_read_lock();
   hello-18623   3.... 1654799.238869: funcgraph_entry:        0.034 us   |        module_put();
   hello-18623   3.... 1654799.238869: funcgraph_exit:       ! 217.327 us |      }
   hello-18623   3.... 1654799.238870: funcgraph_entry:        0.066 us   |      proc_exec_connector();
   hello-18623   3.... 1654799.238870: funcgraph_entry:                   |      acct_update_integrals() {
   hello-18623   3d... 1654799.238870: funcgraph_entry:        0.042 us   |        __acct_update_integrals();
   hello-18623   3.... 1654799.238870: funcgraph_exit:         0.336 us   |      }
   hello-18623   3.... 1654799.238871: funcgraph_entry:                   |      task_numa_free() {
   hello-18623   3.... 1654799.238871: funcgraph_entry:        0.040 us   |        kfree();
   hello-18623   3d... 1654799.238872: funcgraph_entry:        3.233 us   |        smp_irq_work_interrupt();
   hello-18623   3.... 1654799.238876: funcgraph_exit:         5.134 us   |      }
```

# Easier to understand

```
   hello-18623   3.... 1654799.238628: funcgraph_entry:                  |      copy_strings_kernel() {
   hello-18623   3.... 1654799.238628: funcgraph_entry:      + 10.840 us  |        copy_strings.isra.26();
   hello-18623   3.... 1654799.238639: funcgraph_exit:       + 11.232 us  |      }
   hello-18623   3.... 1654799.238640: funcgraph_entry:                  |      copy_strings.isra.26() {
   hello-18623   3.... 1654799.238640: funcgraph_entry:        0.549 us   |        get_user_pages_remote();
   hello-18623   3.... 1654799.238641: funcgraph_entry:        0.090 us   |        __check_object_size();
[..]
   hello-18623   3.... 1654799.238648: funcgraph_entry:        0.085 us   |        __check_object_size();
   hello-18623   3.... 1654799.238649: funcgraph_exit:         8.993 us   |      }
   hello-18623   3.... 1654799.238649: funcgraph_entry:                  |      copy_strings.isra.26() {
   hello-18623   3.... 1654799.238649: funcgraph_entry:        0.449 us   |        get_user_pages_remote();
   hello-18623   3.... 1654799.238650: funcgraph_entry:        0.089 us   |        __check_object_size();
   hello-18623   3.... 1654799.238650: funcgraph_exit:         1.444 us   |      }
   hello-18623   3.... 1654799.238650: funcgraph_entry:                  |      would_dump() {
   hello-18623   3.... 1654799.238651: funcgraph_entry:        0.119 us   |        inode_permission();
   hello-18623   3.... 1654799.238651: funcgraph_exit:         0.415 us   |      }
   hello-18623   3.... 1654799.238651: funcgraph_entry:        0.114 us   |      task_active_pid_ns();
   hello-18623   3.... 1654799.238652: funcgraph_entry:        0.079 us   |      __task_pid_nr_ns();
   hello-18623   3.... 1654799.238652: funcgraph_entry:                  |      search_binary_handler() {
   hello-18623   3.... 1654799.238652: funcgraph_entry:        0.048 us   |        security_bprm_check();
   hello-18623   3.... 1654799.238652: funcgraph_entry:        0.057 us   |        _raw_read_lock();
   hello-18623   3.... 1654799.238653: funcgraph_entry:        0.036 us   |        try_module_get();
   hello-18623   3.... 1654799.238653: funcgraph_entry:      ! 214.953 us |        load_script();
   hello-18623   3.... 1654799.238869: funcgraph_entry:        0.052 us   |        _raw_read_lock();
   hello-18623   3.... 1654799.238869: funcgraph_entry:        0.034 us   |        module_put();
   hello-18623   3.... 1654799.238869: funcgraph_exit:       ! 217.327 us |      }
   hello-18623   3.... 1654799.238870: funcgraph_entry:        0.066 us   |      proc_exec_connector();
   hello-18623   3.... 1654799.238870: funcgraph_entry:                  |      acct_update_integrals() {
   hello-18623   3d... 1654799.238870: funcgraph_entry:        0.042 us   |        __acct_update_integrals();
   hello-18623   3.... 1654799.238870: funcgraph_exit:         0.336 us   |      }
   hello-18623   3.... 1654799.238871: funcgraph_entry:                  |      task_numa_free() {
   hello-18623   3.... 1654799.238871: funcgraph_entry:        0.040 us   |        kfree();
   hello-18623   3d... 1654799.238872: funcgraph_entry:        3.233 us   |        smp_irq_work_interrupt();
   hello-18623   3.... 1654799.238876: funcgraph_exit:         5.134 us   |      }
```

## What’s this load_script doing?

```
# trace-cmd record -p function_graph -g ‘load_script’ --max-graph-depth 3 \
  -n _cond_resched ./hello

  plugin 'function_graph'
hello world!
CPU0 data recorded at offset=0x652000
    0 bytes in size
CPU1 data recorded at offset=0x652000
    0 bytes in size
CPU2 data recorded at offset=0x652000
    0 bytes in size
CPU3 data recorded at offset=0x652000
    4096 bytes in size
```

## Easier to understand

```
# trace-cmd report -IS

CPU 0 is empty
CPU 1 is empty
CPU 2 is empty
cpus=4
   hello-20414   3.... 1656091.711043: funcgraph_entry:                   |  load_script() {
   hello-20414   3.... 1656091.711046: funcgraph_entry:                   |    fput() {
   hello-20414   3.... 1656091.711046: funcgraph_entry:        0.143 us   |      task_work_add();
   hello-20414   3.... 1656091.711047: funcgraph_exit:         0.630 us   |    }
   hello-20414   3.... 1656091.711047: funcgraph_entry:                   |    remove_arg_zero() {
   hello-20414   3.... 1656091.711047: funcgraph_entry:        0.625 us   |      get_user_pages_remote();
   hello-20414   3.... 1656091.711048: funcgraph_exit:         0.971 us   |    }
   hello-20414   3.... 1656091.711048: funcgraph_entry:                   |    copy_strings_kernel() {
   hello-20414   3.... 1656091.711049: funcgraph_entry:        0.844 us   |      copy_strings.isra.26();
   hello-20414   3.... 1656091.711050: funcgraph_exit:         1.146 us   |    }
   hello-20414   3.... 1656091.711050: funcgraph_entry:                   |    copy_strings_kernel() {
   hello-20414   3.... 1656091.711050: funcgraph_entry:        0.769 us   |      copy_strings.isra.26();
   hello-20414   3.... 1656091.711051: funcgraph_exit:         1.070 us   |    }
   hello-20414   3.... 1656091.711051: funcgraph_entry:                   |    bprm_change_interp() {
   hello-20414   3.... 1656091.711051: funcgraph_entry:        0.700 us   |      kstrdup();
   hello-20414   3.... 1656091.711052: funcgraph_exit:         1.107 us   |    }
   hello-20414   3.... 1656091.711053: funcgraph_entry:                   |    open_exec() {
   hello-20414   3.... 1656091.711053: funcgraph_entry:        0.397 us   |      getname_kernel();
   hello-20414   3.... 1656091.711053: funcgraph_entry:        4.998 us   |      do_open_execat();
   hello-20414   3.... 1656091.711059: funcgraph_entry:        0.256 us   |      putname();
   hello-20414   3.... 1656091.711059: funcgraph_exit:         6.559 us   |    }
   hello-20414   3.... 1656091.711059: funcgraph_entry:                   |    prepare_binprm() {
   hello-20414   3.... 1656091.711060: funcgraph_entry:        0.061 us   |      mnt_may_suid();
   hello-20414   3.... 1656091.711060: funcgraph_entry:        1.196 us   |      security_bprm_set_creds();
   hello-20414   3.... 1656091.711061: funcgraph_entry:        1.799 us   |      kernel_read();
   hello-20414   3.... 1656091.711063: funcgraph_exit:         3.909 us   |    }
   hello-20414   3.... 1656091.711064: funcgraph_entry:                   |    search_binary_handler() {
   hello-20414   3.... 1656091.711064: funcgraph_entry:        0.043 us   |      security_bprm_check();
   hello-20414   3.... 1656091.711064: funcgraph_entry:        0.036 us   |      _raw_read_lock();
   hello-20414   3.... 1656091.711064: funcgraph_entry:        0.034 us   |      try_module_get();
   hello-20414   3.... 1656091.711065: funcgraph_entry:        0.050 us   |      load_script();
   hello-20414   3.... 1656091.711065: funcgraph_entry:        0.033 us   |      _raw_read_lock();
   hello-20414   3.... 1656091.711065: funcgraph_entry:        0.035 us   |      module_put();
   hello-20414   3.... 1656091.711066: funcgraph_entry:        0.033 us   |      try_module_get();
   hello-20414   3.... 1656091.711066: funcgraph_entry:      ! 248.708 us |      load_elf_binary();
   hello-20414   3.... 1656091.711316: funcgraph_entry:        0.063 us   |      _raw_read_lock();
   hello-20414   3.... 1656091.711316: funcgraph_entry:        0.036 us   |      module_put();
   hello-20414   3.... 1656091.711317: funcgraph_exit:       ! 253.080 us |    }
   hello-20414   3.... 1656091.711317: funcgraph_exit:       ! 271.034 us |  }
```

## Easier to understand

```
# trace-cmd report -IS

CPU 0 is empty
CPU 1 is empty
CPU 2 is empty
cpus=4
   hello-20414   3.... 1656091.711043: funcgraph_entry:                   |  load_script() {
   hello-20414   3.... 1656091.711046: funcgraph_entry:                   |    fput() {
   hello-20414   3.... 1656091.711046: funcgraph_entry:        0.143 us   |      task_work_add();
   hello-20414   3.... 1656091.711047: funcgraph_exit:         0.630 us   |    }
   hello-20414   3.... 1656091.711047: funcgraph_entry:                   |    remove_arg_zero() {
   hello-20414   3.... 1656091.711047: funcgraph_entry:        0.625 us   |      get_user_pages_remote();
   hello-20414   3.... 1656091.711048: funcgraph_exit:         0.971 us   |    }
   hello-20414   3.... 1656091.711048: funcgraph_entry:                   |    copy_strings_kernel() {
   hello-20414   3.... 1656091.711049: funcgraph_entry:        0.844 us   |      copy_strings.isra.26();
   hello-20414   3.... 1656091.711050: funcgraph_exit:         1.146 us   |    }
   hello-20414   3.... 1656091.711050: funcgraph_entry:                   |    copy_strings_kernel() {
   hello-20414   3.... 1656091.711050: funcgraph_entry:        0.769 us   |      copy_strings.isra.26();
   hello-20414   3.... 1656091.711051: funcgraph_exit:         1.070 us   |    }
   hello-20414   3.... 1656091.711051: funcgraph_entry:                   |    bprm_change_interp() {
   hello-20414   3.... 1656091.711051: funcgraph_entry:        0.700 us   |      kstrdup();
   hello-20414   3.... 1656091.711052: funcgraph_exit:         1.107 us   |    }
   hello-20414   3.... 1656091.711053: funcgraph_entry:                   |    open_exec() {
   hello-20414   3.... 1656091.711053: funcgraph_entry:        0.397 us   |      getname_kernel();
   hello-20414   3.... 1656091.711053: funcgraph_entry:        4.998 us   |      do_open_execat();
   hello-20414   3.... 1656091.711059: funcgraph_entry:        0.256 us   |      putname();
   hello-20414   3.... 1656091.711059: funcgraph_exit:         6.559 us   |    }
   hello-20414   3.... 1656091.711059: funcgraph_entry:                   |    prepare_binprm() {
   hello-20414   3.... 1656091.711060: funcgraph_entry:        0.061 us   |      mnt_may_suid();
   hello-20414   3.... 1656091.711060: funcgraph_entry:        1.196 us   |      security_bprm_set_creds();
   hello-20414   3.... 1656091.711061: funcgraph_entry:        1.799 us   |      kernel_read();
   hello-20414   3.... 1656091.711063: funcgraph_exit:         3.909 us   |    }
   hello-20414   3.... 1656091.711064: funcgraph_entry:                   |    search_binary_handler() {
   hello-20414   3.... 1656091.711064: funcgraph_entry:        0.043 us   |      security_bprm_check();
   hello-20414   3.... 1656091.711064: funcgraph_entry:        0.036 us   |      _raw_read_lock();
   hello-20414   3.... 1656091.711064: funcgraph_entry:        0.034 us   |      try_module_get();
   hello-20414   3.... 1656091.711065: funcgraph_entry:        0.050 us   |      load_script();
   hello-20414   3.... 1656091.711065: funcgraph_entry:        0.033 us   |      _raw_read_lock();
   hello-20414   3.... 1656091.711065: funcgraph_entry:        0.035 us   |      module_put();
   hello-20414   3.... 1656091.711066: funcgraph_entry:        0.033 us   |      try_module_get();
   hello-20414   3.... 1656091.711066: funcgraph_entry:      ! 248.708 us |      load_elf_binary();
   hello-20414   3.... 1656091.711316: funcgraph_entry:        0.063 us   |      _raw_read_lock();
   hello-20414   3.... 1656091.711316: funcgraph_entry:        0.036 us   |      module_put();
   hello-20414   3.... 1656091.711317: funcgraph_exit:       ! 253.080 us |    }
   hello-20414   3.... 1656091.711317: funcgraph_exit:       ! 271.034 us |  }
```

## Recursive function in the kernel?

The kernel has a fixed stack

Recursive functions can blow that stack

Is this a bug?

Can we exploit it?

What is it doing?

## load_script()

```
static int load_script(struct linux_binprm *bprm)
{
     const char *i_arg, *i_name;
     char *cp;
     struct file *file;
     int retval;

     if ((bprm->buf[0] != '#') || (bprm->buf[1] != '!'))
          return -ENOEXEC;
[..]

     /*
      * OK, now restart the process with the interpreter's dentry.
      */
     file = open_exec(i_name);
     if (IS_ERR(file))
          return PTR_ERR(file);

     bprm->file = file;
     retval = prepare_binprm(bprm);
     if (retval < 0)
          return retval;
     return search_binary_handler(bprm);
}
```

## Can we exploit this?

```
#!/tmp/blah

Crash me!
```

```
# echo '#!/tmp/blah

Crash me!' > /tmp/blah

# chmod +x /tmp/blah
# /tmp/blah
```

## Can we exploit this? Nope!

```
#!/tmp/blah

Crash me!
```

```
# echo ‘#!/tmp/blah

Crash me!’ > /tmp/blah

# chmod +x /tmp/blah
# /tmp/blah

bash: /tmp/blah: /tmp/blah: bad interpreter: Too many levels of symbolic links
```

## Let’s see what happened

```
# trace-cmd record -p function_graph -g ‘load_script’ sh -c /tmp/blah

  plugin 'function_graph'
sh: 1: /tmp/blah: Too many levels of symbolic links
CPU0 data recorded at offset=0x652000
    0 bytes in size
CPU1 data recorded at offset=0x652000
    155648 bytes in size
CPU2 data recorded at offset=0x678000
    4096 bytes in size
CPU3 data recorded at offset=0x679000
    0 bytes in size
```

## Follow the load_scripts

```
# trace-cmd report -O tailprint | grep load_script

            sh-28473 [002] 1660671.155766: funcgraph_entry:        0.122 us   |  load_script();
            sh-28474 [001] 1660671.156664: funcgraph_entry:                   |  load_script() {
            sh-28474 [001] 1660671.156822: funcgraph_entry:                   |    load_script() {
            sh-28474 [001] 1660671.156999: funcgraph_entry:                   |      load_script() {
            sh-28474 [001] 1660671.157137: funcgraph_entry:                   |        load_script() {
            sh-28474 [001] 1660671.157248: funcgraph_entry:                   |          load_script() {
            sh-28474 [001] 1660671.157361: funcgraph_entry:                   |            load_script() {
            sh-28474 [001] 1660671.157449: funcgraph_exit:       + 88.202 us  |            } /* load_script */
            sh-28474 [001] 1660671.157450: funcgraph_exit:       ! 201.306 us |          } /* load_script */
            sh-28474 [001] 1660671.157451: funcgraph_exit:       ! 313.092 us |        } /* load_script */
            sh-28474 [001] 1660671.157451: funcgraph_exit:       ! 452.196 us |      } /* load_script */
            sh-28474 [001] 1660671.157452: funcgraph_exit:       ! 630.448 us |    } /* load_script */
            sh-28474 [001] 1660671.157453: funcgraph_exit:       ! 770.883 us |  } /* load_script */
```

## Follow the load_scripts and a little more

```
# trace-cmd report -O tailprint | grep -C 2 load_script

CPU 3 is empty
cpus=4
              sh-28473 [002] 1660671.155766: funcgraph_entry:        0.122 us   |  load_script();
              sh-28474 [001] 1660671.156664: funcgraph_entry:                   |  load_script() {
              sh-28474 [001] 1660671.156682: funcgraph_entry:                   |    fput() {
              sh-28474 [001] 1660671.156683: funcgraph_entry:                   |      task_work_add() {
--
              sh-28474 [001] 1660671.156821: funcgraph_entry:        0.043 us   |      _raw_read_lock();
              sh-28474 [001] 1660671.156821: funcgraph_entry:        0.031 us   |      try_module_get();
              sh-28474 [001] 1660671.156822: funcgraph_entry:                   |      load_script() {
              sh-28474 [001] 1660671.156822: funcgraph_entry:                   |        fput() {
              sh-28474 [001] 1660671.156822: funcgraph_entry:        0.056 us   |          task_work_add();
--
              sh-28474 [001] 1660671.156999: funcgraph_entry:        0.039 us   |          _raw_read_lock();
              sh-28474 [001] 1660671.156999: funcgraph_entry:        0.032 us   |          try_module_get();
              sh-28474 [001] 1660671.156999: funcgraph_entry:                   |          load_script() {
              sh-28474 [001] 1660671.156999: funcgraph_entry:                   |            fput() {
              sh-28474 [001] 1660671.156999: funcgraph_entry:        0.048 us   |              task_work_add();
--
              sh-28474 [001] 1660671.157137: funcgraph_entry:        0.033 us   |              _raw_read_lock();
              sh-28474 [001] 1660671.157137: funcgraph_entry:        0.032 us   |              try_module_get();
              sh-28474 [001] 1660671.157137: funcgraph_entry:                   |              load_script() {
              sh-28474 [001] 1660671.157138: funcgraph_entry:                   |                fput() {
              sh-28474 [001] 1660671.157138: funcgraph_entry:        0.052 us   |                  task_work_add();
--
              sh-28474 [001] 1660671.157248: funcgraph_entry:        0.032 us   |                  _raw_read_lock();
              sh-28474 [001] 1660671.157248: funcgraph_entry:        0.032 us   |                  try_module_get();
              sh-28474 [001] 1660671.157248: funcgraph_entry:                   |                  load_script() {
              sh-28474 [001] 1660671.157249: funcgraph_entry:                   |                    fput() {
              sh-28474 [001] 1660671.157249: funcgraph_entry:        0.050 us   |                      task_work_add();
--
              sh-28474 [001] 1660671.157360: funcgraph_exit:         0.037 us   |                      } /* _raw_read_lock */
              sh-28474 [001] 1660671.157360: funcgraph_entry:        0.032 us   |                      try_module_get();
              sh-28474 [001] 1660671.157361: funcgraph_entry:                   |                      load_script() {
              sh-28474 [001] 1660671.157361: funcgraph_entry:                   |                        fput() {
              sh-28474 [001] 1660671.157361: funcgraph_entry:        0.040 us   |                          task_work_add();
--
              sh-28474 [001] 1660671.157449: funcgraph_exit:       + 15.922 us  |                        } /* prepare_binprm */
              sh-28474 [001] 1660671.157449: funcgraph_entry:        0.054 us   |                        search_binary_handler();
              sh-28474 [001] 1660671.157449: funcgraph_exit:       + 88.202 us  |                      } /* load_script */
```

# Follow the load_scripts and a little more

```
# trace-cmd report -O tailprint | grep -C 2 load_script

CPU 3 is empty
cpus=4
              sh-28473 [002] 1660671.155766: funcgraph_entry:        0.122 us   |  load_script();
              sh-28474 [001] 1660671.156664: funcgraph_entry:                   |  load_script() {
              sh-28474 [001] 1660671.156682: funcgraph_entry:                   |    fput() {
              sh-28474 [001] 1660671.156683: funcgraph_entry:                   |      task_work_add() {
--
              sh-28474 [001] 1660671.156821: funcgraph_entry:        0.043 us   |      _raw_read_lock();
              sh-28474 [001] 1660671.156821: funcgraph_entry:        0.031 us   |      try_module_get();
              sh-28474 [001] 1660671.156822: funcgraph_entry:                   |      load_script() {
              sh-28474 [001] 1660671.156822: funcgraph_entry:                   |        fput() {
              sh-28474 [001] 1660671.156822: funcgraph_entry:        0.056 us   |          task_work_add();
--
              sh-28474 [001] 1660671.156999: funcgraph_entry:        0.039 us   |          _raw_read_lock();
              sh-28474 [001] 1660671.156999: funcgraph_entry:        0.032 us   |          try_module_get();
              sh-28474 [001] 1660671.156999: funcgraph_entry:                   |          load_script() {
              sh-28474 [001] 1660671.156999: funcgraph_entry:                   |            fput() {
              sh-28474 [001] 1660671.156999: funcgraph_entry:        0.048 us   |              task_work_add();
--
              sh-28474 [001] 1660671.157137: funcgraph_entry:        0.033 us   |              _raw_read_lock();
              sh-28474 [001] 1660671.157137: funcgraph_entry:        0.032 us   |              try_module_get();
              sh-28474 [001] 1660671.157137: funcgraph_entry:                   |              load_script() {
              sh-28474 [001] 1660671.157138: funcgraph_entry:                   |                fput() {
              sh-28474 [001] 1660671.157138: funcgraph_entry:        0.052 us   |                  task_work_add();
--
              sh-28474 [001] 1660671.157248: funcgraph_entry:        0.032 us   |                  _raw_read_lock();
              sh-28474 [001] 1660671.157248: funcgraph_entry:        0.032 us   |                  try_module_get();
              sh-28474 [001] 1660671.157248: funcgraph_entry:                   |                  load_script() {
              sh-28474 [001] 1660671.157249: funcgraph_entry:                   |                    fput() {
              sh-28474 [001] 1660671.157249: funcgraph_entry:        0.050 us   |                      task_work_add();
--
              sh-28474 [001] 1660671.157360: funcgraph_exit:         0.037 us   |                      } /* _raw_read_lock */
              sh-28474 [001] 1660671.157360: funcgraph_entry:        0.032 us   |                      try_module_get();
              sh-28474 [001] 1660671.157361: funcgraph_entry:                   |                      load_script() {
              sh-28474 [001] 1660671.157361: funcgraph_entry:                   |                        fput() {
              sh-28474 [001] 1660671.157361: funcgraph_entry:        0.040 us   |                          task_work_add();
--
              sh-28474 [001] 1660671.157449: funcgraph_exit:       + 15.922 us  |                        } /* prepare_binprm */
              sh-28474 [001] 1660671.157449: funcgraph_entry:        0.054 us   |                        search_binary_handler();
              sh-28474 [001] 1660671.157449: funcgraph_exit:       + 88.202 us  |                      } /* load_script */
```

# search_binary_handler()

```
/*
 * cycle the list of binary formats handler, until one recognizes the image
 */
int search_binary_handler(struct linux_binprm *bprm)
{
     bool need_retry = IS_ENABLED(CONFIG_MODULES);
     struct linux_binfmt *fmt;
     int retval;

     /* This allows 4 levels of binfmt rewrites before failing hard. */
     if (bprm->recursion_depth > 5)
          return -ELOOP;
```

## I want to know what my computer is doing!

OK, that was interesting (and time consuming)

But what about the rest of my computer?

I want to see it all?

Well, perhaps not all, function tracing is a bit overwhelming!

Just the events please

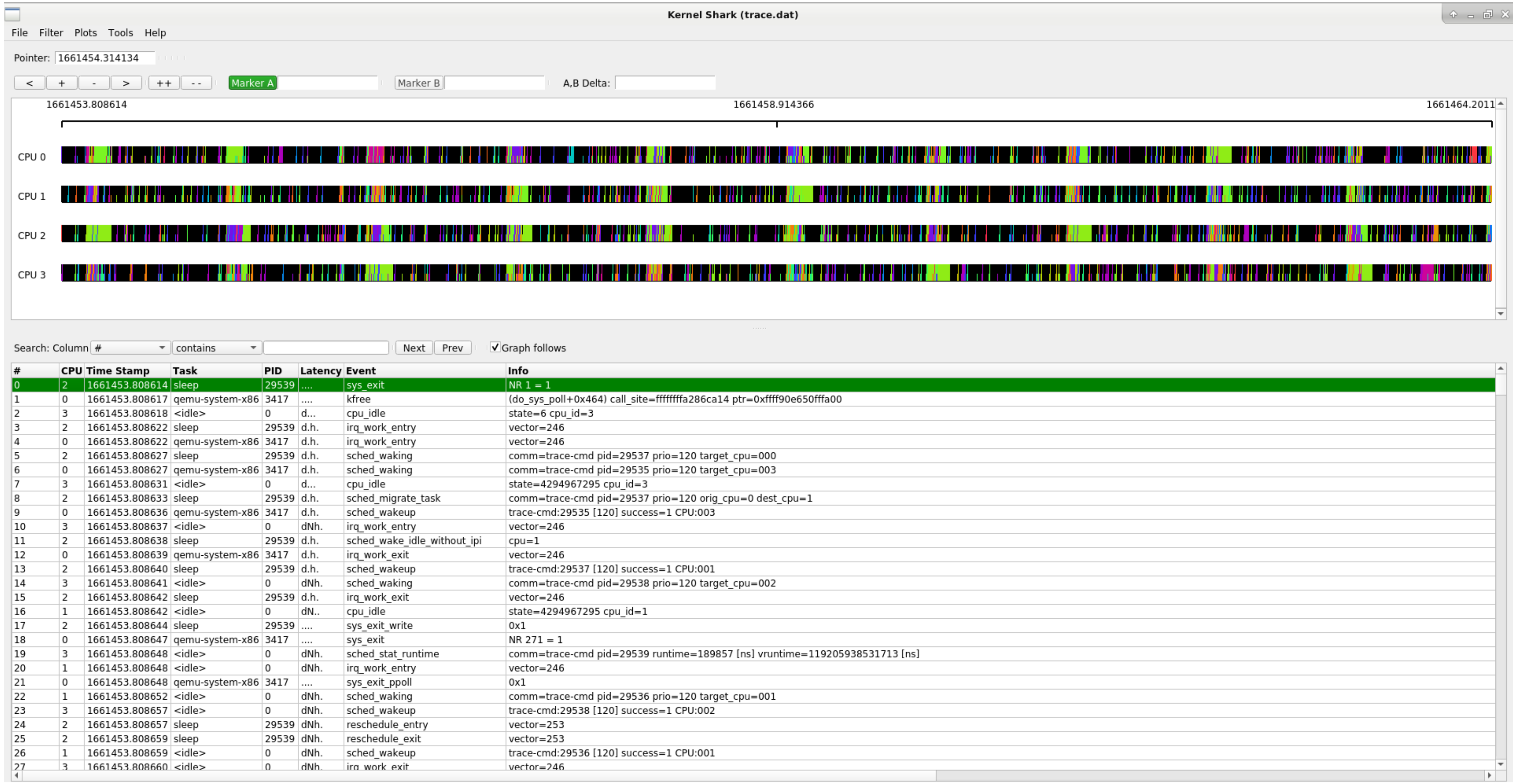
## Let’s see the world!

```
# trace-cmd record -e all sleep 10

CPU0 data recorded at offset=0x822000
    20647936 bytes in size
CPU1 data recorded at offset=0x1bd3000
    21708800 bytes in size
CPU2 data recorded at offset=0x3087000
    20733952 bytes in size
CPU3 data recorded at offset=0x444d000
    22446080 bytes in size
```

# Well, um, that’s informative. I think?

```
# trace-cmd report

cpus=4
           sleep-29539 [002] 1661453.808614: sys_exit:             NR 1 = 1
 qemu-system-x86-3417  [000] 1661453.808617: kfree:                (do_sys_poll+0x464) call_site=ffffffffa286ca14 ptr=0xffff90e650fffa00
          <idle>-0     [003] 1661453.808618: cpu_idle:             state=6 cpu_id=3
           sleep-29539 [002] 1661453.808622: irq_work_entry:       vector=246
 qemu-system-x86-3417  [000] 1661453.808622: irq_work_entry:       vector=246
           sleep-29539 [002] 1661453.808627: sched_waking:         comm=trace-cmd pid=29537 prio=120 target_cpu=000
 qemu-system-x86-3417  [000] 1661453.808627: sched_waking:         comm=trace-cmd pid=29535 prio=120 target_cpu=003
          <idle>-0     [003] 1661453.808631: cpu_idle:             state=4294967295 cpu_id=3
           sleep-29539 [002] 1661453.808633: sched_migrate_task:   comm=trace-cmd pid=29537 prio=120 orig_cpu=0 dest_cpu=1
 qemu-system-x86-3417  [000] 1661453.808636: sched_wakeup:         trace-cmd:29535 [120] success=1 CPU:003
          <idle>-0     [003] 1661453.808637: irq_work_entry:       vector=246
           sleep-29539 [002] 1661453.808638: sched_wake_idle_without_ipi: cpu=1
 qemu-system-x86-3417  [000] 1661453.808639: irq_work_exit:        vector=246
           sleep-29539 [002] 1661453.808640: sched_wakeup:         trace-cmd:29537 [120] success=1 CPU:001
          <idle>-0     [003] 1661453.808641: sched_waking:         comm=trace-cmd pid=29538 prio=120 target_cpu=002
           sleep-29539 [002] 1661453.808642: irq_work_exit:        vector=246
          <idle>-0     [001] 1661453.808642: cpu_idle:             state=4294967295 cpu_id=1
           sleep-29539 [002] 1661453.808644: sys_exit_write:       0x1
 qemu-system-x86-3417  [000] 1661453.808647: sys_exit:             NR 271 = 1
          <idle>-0     [003] 1661453.808648: sched_stat_runtime:   comm=trace-cmd pid=29539 runtime=189857 [ns] vruntime=119205938531713 [ns]
          <idle>-0     [001] 1661453.808648: irq_work_entry:       vector=246
 qemu-system-x86-3417  [000] 1661453.808648: sys_exit_ppoll:       0x1
          <idle>-0     [001] 1661453.808652: sched_waking:         comm=trace-cmd pid=29536 prio=120 target_cpu=001
          <idle>-0     [003] 1661453.808657: sched_wakeup:         trace-cmd:29538 [120] success=1 CPU:002
           sleep-29539 [002] 1661453.808657: reschedule_entry:     vector=253
           sleep-29539 [002] 1661453.808659: reschedule_exit:      vector=253
          <idle>-0     [001] 1661453.808659: sched_wakeup:         trace-cmd:29536 [120] success=1 CPU:001
          <idle>-0     [003] 1661453.808660: irq_work_exit:        vector=246
          <idle>-0     [001] 1661453.808661: irq_work_exit:        vector=246
           sleep-29539 [002] 1661453.808661: rcu_utilization:      Start context switch
           sleep-29539 [002] 1661453.808663: rcu_utilization:      End context switch
          <idle>-0     [003] 1661453.808664: reschedule_entry:     vector=253
          <idle>-0     [001] 1661453.808664: rcu_utilization:      Start context switch
          <idle>-0     [003] 1661453.808666: reschedule_exit:      vector=253
          <idle>-0     [001] 1661453.808666: rcu_utilization:      End context switch
          <idle>-0     [003] 1661453.808668: rcu_utilization:      Start context switch
           sleep-29539 [002] 1661453.808669: sched_switch:         trace-cmd:29539 [120] S ==> trace-cmd:29538 [120]
          <idle>-0     [003] 1661453.808670: rcu_utilization:      End context switch
          <idle>-0     [001] 1661453.808670: sched_switch:         swapper/1:0 [120] S ==> trace-cmd:29537 [120]
```

# What about this?



| # | CPU | Time Stamp | Task | PID | Latency | Event | Info |
|---|---|---|---|---|---|---|---|
| 0 | 2 | 1661453.808614 | sleep | 29539 | .... | sys_exit | NR 1 = 1 |
| 1 | 0 | 1661453.808617 | qemu-system-x86 | 3417 | .... | kfree | (do_sys_poll+0x464) call_site=ffffffffa286ca14 ptr=0xffff90e650fffa00 |
| 2 | 3 | 1661453.808618 | <idle> | 0 | d... | cpu_idle | state=6 cpu_id=3 |
| 3 | 2 | 1661453.808622 | sleep | 29539 | d.h. | irq_work_entry | vector=246 |
| 4 | 0 | 1661453.808622 | qemu-system-x86 | 3417 | d.h. | irq_work_entry | vector=246 |
| 5 | 2 | 1661453.808627 | sleep | 29539 | d.h. | sched_waking | comm=trace-cmd pid=29537 prio=120 target_cpu=000 |
| 6 | 0 | 1661453.808627 | qemu-system-x86 | 3417 | d.h. | sched_waking | comm=trace-cmd pid=29535 prio=120 target_cpu=003 |
| 7 | 3 | 1661453.808631 | <idle> | 0 | d... | cpu_idle | state=4294967295 cpu_id=3 |
| 8 | 2 | 1661453.808633 | sleep | 29539 | d.h. | sched_migrate_task | comm=trace-cmd pid=29537 prio=120 orig_cpu=0 dest_cpu=1 |
| 9 | 0 | 1661453.808636 | qemu-system-x86 | 3417 | d.h. | sched_wakeup | trace-cmd:29535 [120] success=1 CPU:003 |
| 10 | 3 | 1661453.808637 | <idle> | 0 | dNh. | irq_work_entry | vector=246 |
| 11 | 2 | 1661453.808638 | sleep | 29539 | d.h. | sched_wake_idle_without_ipi | cpu=1 |
| 12 | 0 | 1661453.808639 | qemu-system-x86 | 3417 | d.h. | irq_work_exit | vector=246 |
| 13 | 2 | 1661453.808640 | sleep | 29539 | d.h. | sched_wakeup | trace-cmd:29537 [120] success=1 CPU:001 |
| 14 | 3 | 1661453.808641 | <idle> | 0 | dNh. | sched_waking | comm=trace-cmd pid=29538 prio=120 target_cpu=002 |
| 15 | 2 | 1661453.808642 | sleep | 29539 | d.h. | irq_work_exit | vector=246 |
| 16 | 1 | 1661453.808642 | <idle> | 0 | dN.. | cpu_idle | state=4294967295 cpu_id=1 |
| 17 | 2 | 1661453.808644 | sleep | 29539 | .... | sys_exit_write | 0x1 |
| 18 | 0 | 1661453.808647 | qemu-system-x86 | 3417 | .... | sys_exit | NR 271 = 1 |
| 19 | 3 | 1661453.808648 | <idle> | 0 | dNh. | sched_stat_runtime | comm=trace-cmd pid=29539 runtime=189857 [ns] vruntime=119205938531713 [ns] |
| 20 | 1 | 1661453.808648 | <idle> | 0 | dNh. | irq_work_entry | vector=246 |
| 21 | 0 | 1661453.808648 | qemu-system-x86 | 3417 | .... | sys_exit_ppoll | 0x1 |
| 22 | 1 | 1661453.808652 | <idle> | 0 | dNh. | sched_waking | comm=trace-cmd pid=29536 prio=120 target_cpu=001 |
| 23 | 3 | 1661453.808657 | <idle> | 0 | dNh. | sched_wakeup | trace-cmd:29538 [120] success=1 CPU:002 |
| 24 | 2 | 1661453.808657 | sleep | 29539 | dNh. | reschedule_entry | vector=253 |
| 25 | 2 | 1661453.808659 | sleep | 29539 | dNh. | reschedule_exit | vector=253 |
| 26 | 1 | 1661453.808659 | <idle> | 0 | dNh. | sched_wakeup | trace-cmd:29536 [120] success=1 CPU:001 |
| 27 | 3 | 1661453.808660 | <idle> | 0 | dNh. | irq_work_exit | vector=246 |

## Introducing KernelShark 1.0

Well, at least it’s prettier

## Introducing KernelShark 1.0

KernelShark was create in 2009

- GTK version
- Stalled in development
- Slow on large data sets

VMware hired someone to work on it full time

- Rewritten from scratch
- Uses Qt
- Handles large data sets quickly
- Is now a platform (ftrace layer is abstracted out)

## Introducing KernelShark 0.99 actually

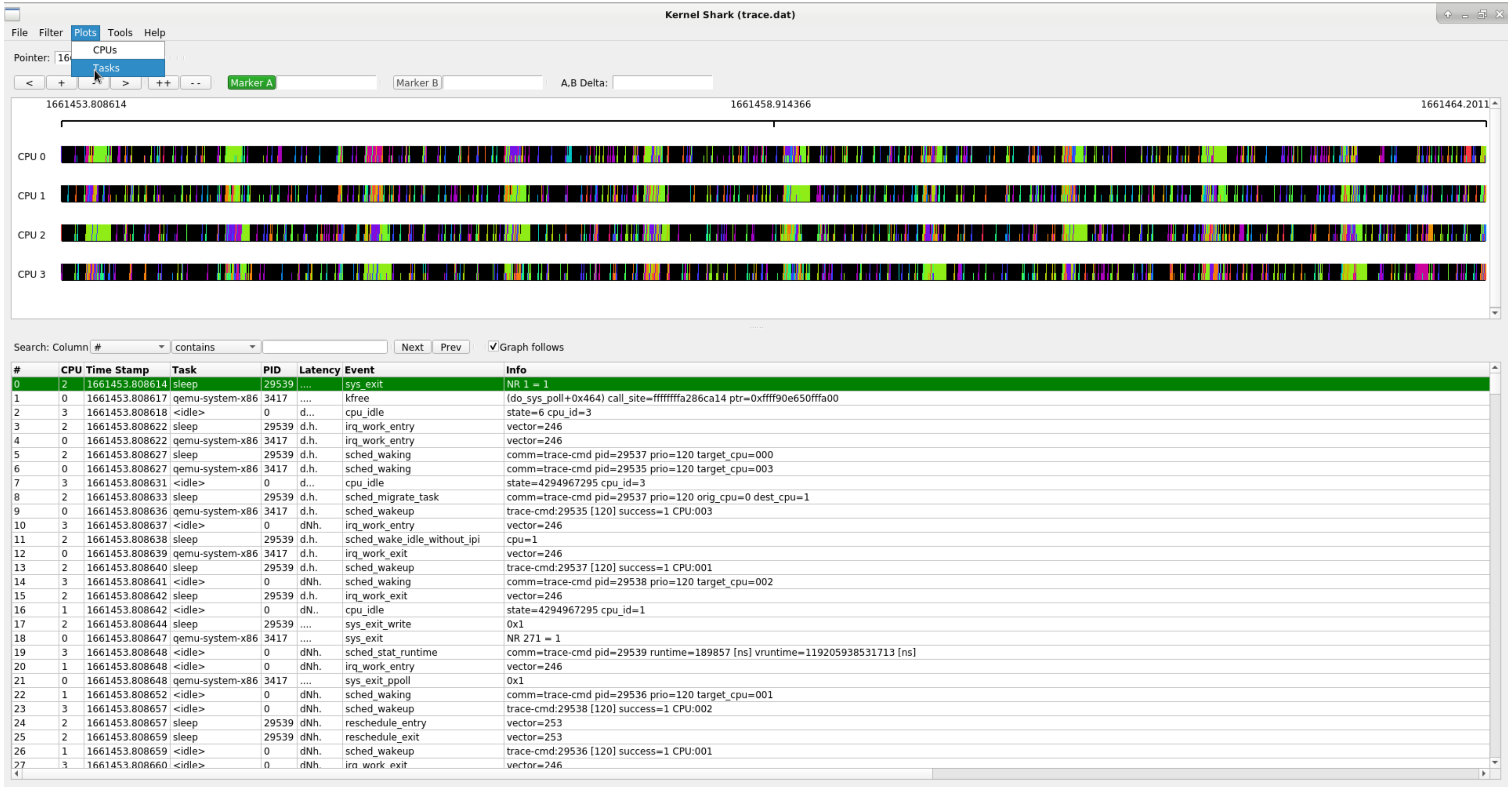
Congratulations!

- You are all now beta testers
- Find a bug, report it here:

  https://bugzilla.kernel.org

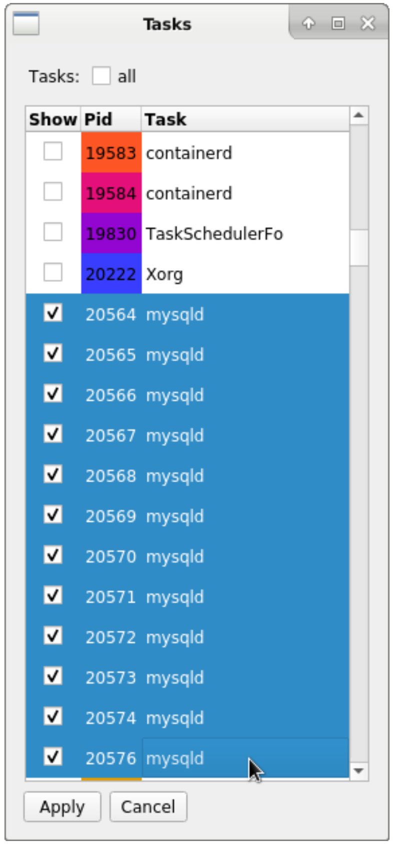
  - Pick “Tools: Tools and utilities”
  - Then “Trace-cmd/Kernelshark”

# KernelShark 1.0



| # | CPU | Time Stamp | Task | PID | Latency | Event | Info |
|---|---|---|---|---|---|---|---|
| 0 | 2 | 1661453.808614 | sleep | 29539 | .... | sys_exit | NR 1 = 1 |
| 1 | 0 | 1661453.808617 | qemu-system-x86 | 3417 | .... | kfree | (do_sys_poll+0x464) call_site=ffffffffa286ca14 ptr=0xffff90e650fffa00 |
| 2 | 3 | 1661453.808618 | <idle> | 0 | d... | cpu_idle | state=6 cpu_id=3 |
| 3 | 2 | 1661453.808622 | sleep | 29539 | d.h. | irq_work_entry | vector=246 |
| 4 | 0 | 1661453.808622 | qemu-system-x86 | 3417 | d.h. | irq_work_entry | vector=246 |
| 5 | 2 | 1661453.808627 | sleep | 29539 | d.h. | sched_waking | comm=trace-cmd pid=29537 prio=120 target_cpu=000 |
| 6 | 0 | 1661453.808627 | qemu-system-x86 | 3417 | d.h. | sched_waking | comm=trace-cmd pid=29535 prio=120 target_cpu=003 |
| 7 | 3 | 1661453.808631 | <idle> | 0 | d... | cpu_idle | state=4294967295 cpu_id=3 |
| 8 | 2 | 1661453.808633 | sleep | 29539 | d.h. | sched_migrate_task | comm=trace-cmd pid=29537 prio=120 orig_cpu=0 dest_cpu=1 |
| 9 | 0 | 1661453.808636 | qemu-system-x86 | 3417 | d.h. | sched_wakeup | trace-cmd:29535 [120] success=1 CPU:003 |
| 10 | 3 | 1661453.808637 | <idle> | 0 | dNh. | irq_work_entry | vector=246 |
| 11 | 2 | 1661453.808638 | sleep | 29539 | d.h. | sched_wake_idle_without_ipi | cpu=1 |
| 12 | 0 | 1661453.808639 | qemu-system-x86 | 3417 | d.h. | irq_work_exit | vector=246 |
| 13 | 2 | 1661453.808640 | sleep | 29539 | d.h. | sched_wakeup | trace-cmd:29537 [120] success=1 CPU:001 |
| 14 | 3 | 1661453.808641 | <idle> | 0 | dNh. | sched_waking | comm=trace-cmd pid=29538 prio=120 target_cpu=002 |
| 15 | 2 | 1661453.808642 | sleep | 29539 | d.h. | irq_work_exit | vector=246 |
| 16 | 1 | 1661453.808642 | <idle> | 0 | dN.. | cpu_idle | state=4294967295 cpu_id=1 |
| 17 | 2 | 1661453.808644 | sleep | 29539 | .... | sys_exit_write | 0x1 |
| 18 | 0 | 1661453.808647 | qemu-system-x86 | 3417 | .... | sys_exit | NR 271 = 1 |
| 19 | 3 | 1661453.808648 | <idle> | 0 | dNh. | sched_stat_runtime | comm=trace-cmd pid=29539 runtime=189857 [ns] vruntime=119205938531713 [ns] |
| 20 | 1 | 1661453.808648 | <idle> | 0 | dNh. | irq_work_entry | vector=246 |
| 21 | 0 | 1661453.808648 | qemu-system-x86 | 3417 | .... | sys_exit_ppoll | 0x1 |
| 22 | 1 | 1661453.808652 | <idle> | 0 | dNh. | sched_waking | comm=trace-cmd pid=29536 prio=120 target_cpu=001 |
| 23 | 3 | 1661453.808657 | <idle> | 0 | dNh. | sched_wakeup | trace-cmd:29538 [120] success=1 CPU:002 |
| 24 | 2 | 1661453.808657 | sleep | 29539 | dNh. | reschedule_entry | vector=253 |
| 25 | 2 | 1661453.808659 | sleep | 29539 | dNh. | reschedule_exit | vector=253 |
| 26 | 1 | 1661453.808659 | <idle> | 0 | dNh. | sched_wakeup | trace-cmd:29536 [120] success=1 CPU:001 |
| 27 | 3 | 1661453.808660 | <idle> | 0 | dNh. | irq_work_exit | vector=246 |

# KernelShark 1.0 (Task Selection)



Tasks: ☐ all

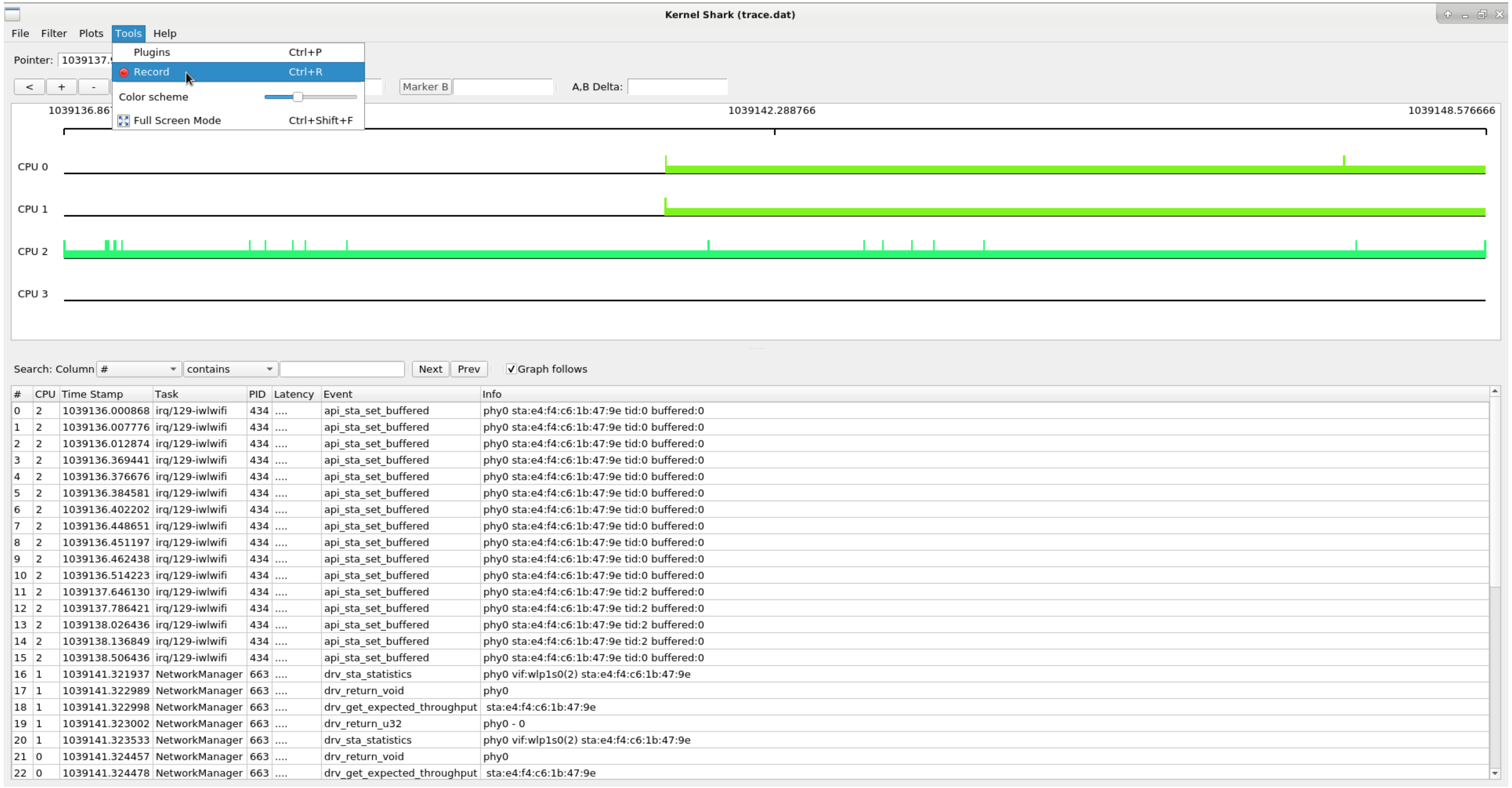
| Show | Pid | Task |
|---|---|---|
| ☐ | 19583 | containerd |
| ☐ | 19584 | containerd |
| ☐ | 19830 | TaskSchedulerFo |
| ☐ | 20222 | Xorg |
| ✓ | 20564 | mysqld |
| ✓ | 20565 | mysqld |
| ✓ | 20566 | mysqld |
| ✓ | 20567 | mysqld |
| ✓ | 20568 | mysqld |
| ✓ | 20569 | mysqld |
| ✓ | 20570 | mysqld |
| ✓ | 20571 | mysqld |
| ✓ | 20572 | mysqld |
| ✓ | 20573 | mysqld |
| ✓ | 20574 | mysqld |
| ✓ | 20576 | mysqld |

## A really simple java program!

Using your favorite editor, create this file

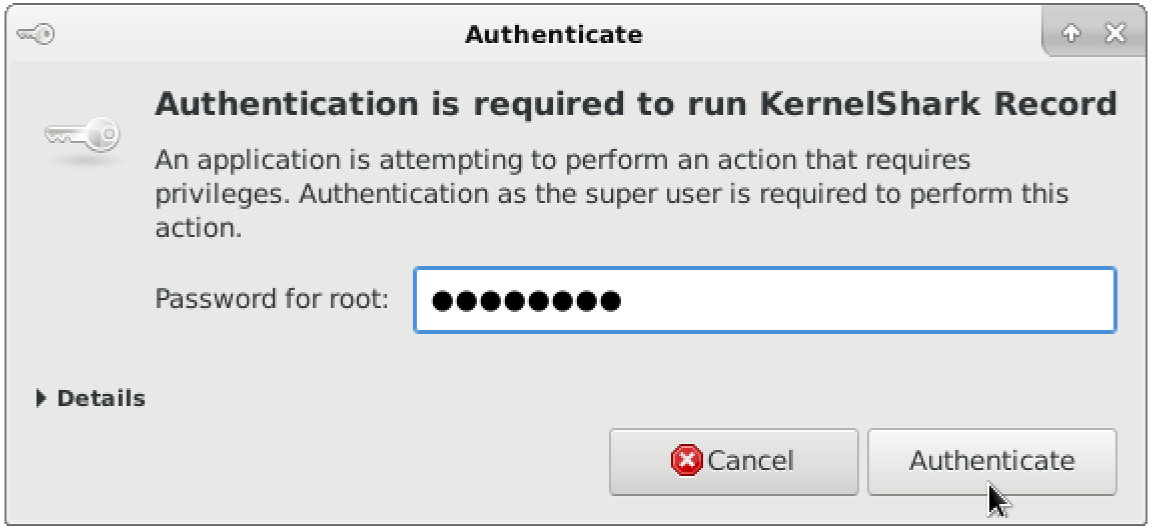
```
public class HelloWorld {

    public static void main(String[] args) {
        System.out.println("Hello, World");
    }

}
```

# KernelShark 1.0 (Recording)



| # | CPU | Time Stamp | Task | PID | Latency | Event | Info |
|---|---|---|---|---|---|---|---|
| 0 | 2 | 1039136.000868 | irq/129-iwlwifi | 434 | .... | api_sta_set_buffered | phy0 sta:e4:f4:c6:1b:47:9e tid:0 buffered:0 |
| 1 | 2 | 1039136.007776 | irq/129-iwlwifi | 434 | .... | api_sta_set_buffered | phy0 sta:e4:f4:c6:1b:47:9e tid:0 buffered:0 |
| 2 | 2 | 1039136.012874 | irq/129-iwlwifi | 434 | .... | api_sta_set_buffered | phy0 sta:e4:f4:c6:1b:47:9e tid:0 buffered:0 |
| 3 | 2 | 1039136.369441 | irq/129-iwlwifi | 434 | .... | api_sta_set_buffered | phy0 sta:e4:f4:c6:1b:47:9e tid:0 buffered:0 |
| 4 | 2 | 1039136.376676 | irq/129-iwlwifi | 434 | .... | api_sta_set_buffered | phy0 sta:e4:f4:c6:1b:47:9e tid:0 buffered:0 |
| 5 | 2 | 1039136.384581 | irq/129-iwlwifi | 434 | .... | api_sta_set_buffered | phy0 sta:e4:f4:c6:1b:47:9e tid:0 buffered:0 |
| 6 | 2 | 1039136.402202 | irq/129-iwlwifi | 434 | .... | api_sta_set_buffered | phy0 sta:e4:f4:c6:1b:47:9e tid:0 buffered:0 |
| 7 | 2 | 1039136.448651 | irq/129-iwlwifi | 434 | .... | api_sta_set_buffered | phy0 sta:e4:f4:c6:1b:47:9e tid:0 buffered:0 |
| 8 | 2 | 1039136.451197 | irq/129-iwlwifi | 434 | .... | api_sta_set_buffered | phy0 sta:e4:f4:c6:1b:47:9e tid:0 buffered:0 |
| 9 | 2 | 1039136.462438 | irq/129-iwlwifi | 434 | .... | api_sta_set_buffered | phy0 sta:e4:f4:c6:1b:47:9e tid:0 buffered:0 |
| 10 | 2 | 1039136.514223 | irq/129-iwlwifi | 434 | .... | api_sta_set_buffered | phy0 sta:e4:f4:c6:1b:47:9e tid:0 buffered:0 |
| 11 | 2 | 1039137.646130 | irq/129-iwlwifi | 434 | .... | api_sta_set_buffered | phy0 sta:e4:f4:c6:1b:47:9e tid:2 buffered:0 |
| 12 | 2 | 1039137.786421 | irq/129-iwlwifi | 434 | .... | api_sta_set_buffered | phy0 sta:e4:f4:c6:1b:47:9e tid:2 buffered:0 |
| 13 | 2 | 1039138.026436 | irq/129-iwlwifi | 434 | .... | api_sta_set_buffered | phy0 sta:e4:f4:c6:1b:47:9e tid:2 buffered:0 |
| 14 | 2 | 1039138.136849 | irq/129-iwlwifi | 434 | .... | api_sta_set_buffered | phy0 sta:e4:f4:c6:1b:47:9e tid:2 buffered:0 |
| 15 | 2 | 1039138.506436 | irq/129-iwlwifi | 434 | .... | api_sta_set_buffered | phy0 sta:e4:f4:c6:1b:47:9e tid:0 buffered:0 |
| 16 | 1 | 1039141.321937 | NetworkManager | 663 | .... | drv_sta_statistics | phy0 vif:wlp1s0(2) sta:e4:f4:c6:1b:47:9e |
| 17 | 1 | 1039141.322989 | NetworkManager | 663 | .... | drv_return_void | phy0 |
| 18 | 1 | 1039141.322998 | NetworkManager | 663 | .... | drv_get_expected_throughput | sta:e4:f4:c6:1b:47:9e |
| 19 | 1 | 1039141.323002 | NetworkManager | 663 | .... | drv_return_u32 | phy0 - 0 |
| 20 | 1 | 1039141.323533 | NetworkManager | 663 | .... | drv_sta_statistics | phy0 vif:wlp1s0(2) sta:e4:f4:c6:1b:47:9e |
| 21 | 0 | 1039141.324457 | NetworkManager | 663 | .... | drv_return_void | phy0 |
| 22 | 0 | 1039141.324478 | NetworkManager | 663 | .... | drv_get_expected_throughput | sta:e4:f4:c6:1b:47:9e |

# KernelShark 1.0 (Recording)



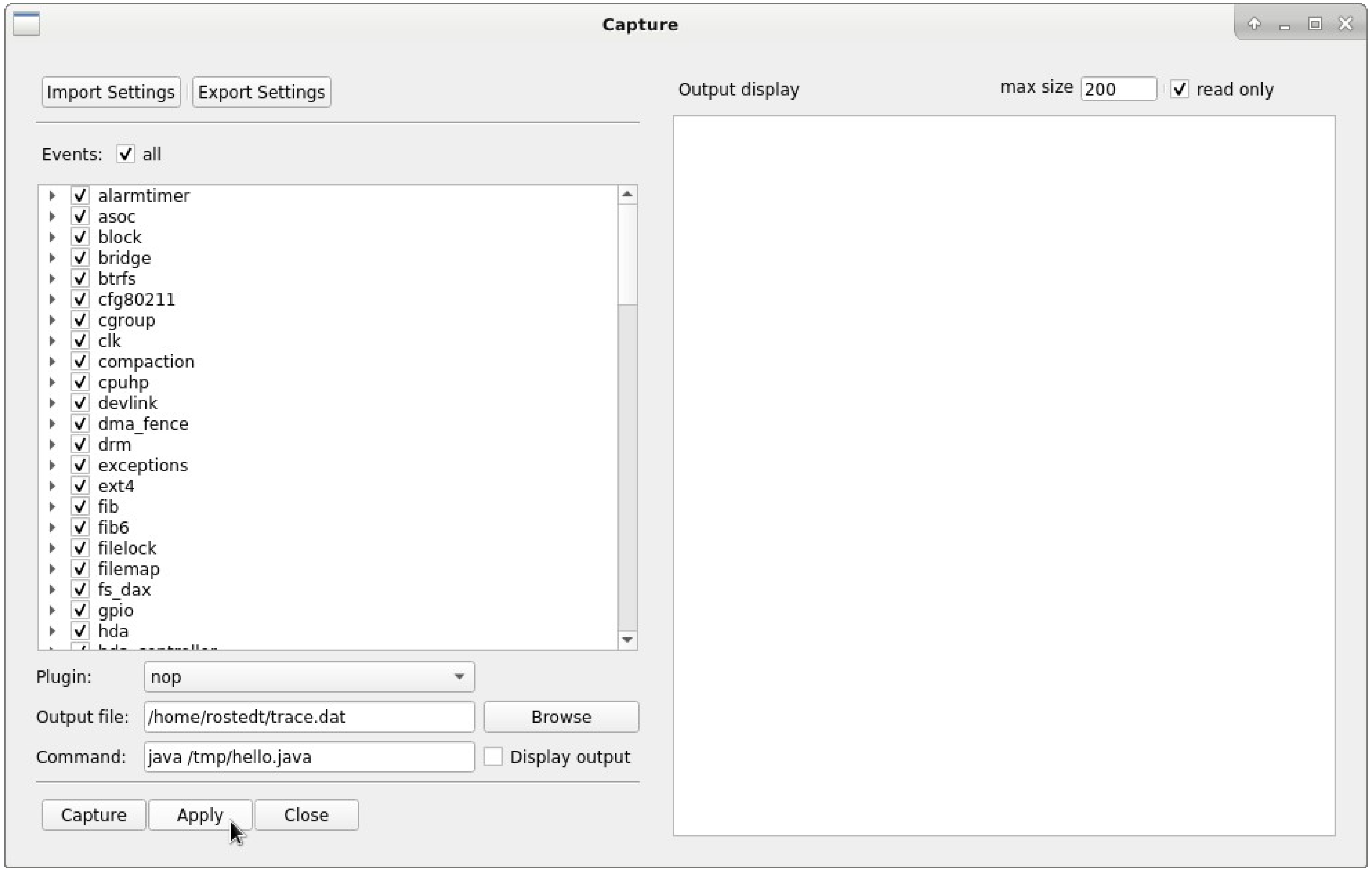
Authenticate

**Authentication is required to run KernelShark Record**

An application is attempting to perform an action that requires privileges. Authentication as the super user is required to perform this action.

Password for root: ●●●●●●●●

# KernelShark 1.0 (Recording)


Capture

Import Settings | Export Settings

Events: ✓ all

✓ alarmtimer
✓ asoc
✓ block
✓ bridge
✓ btrfs
✓ cfg80211
✓ cgroup
✓ clk
✓ compaction
✓ cpuhp
✓ devlink
✓ dma_fence
✓ drm
✓ exceptions
✓ ext4
✓ fib
✓ fib6
✓ filelock
✓ filemap
✓ fs_dax
✓ gpio
✓ hda

Plugin: nop

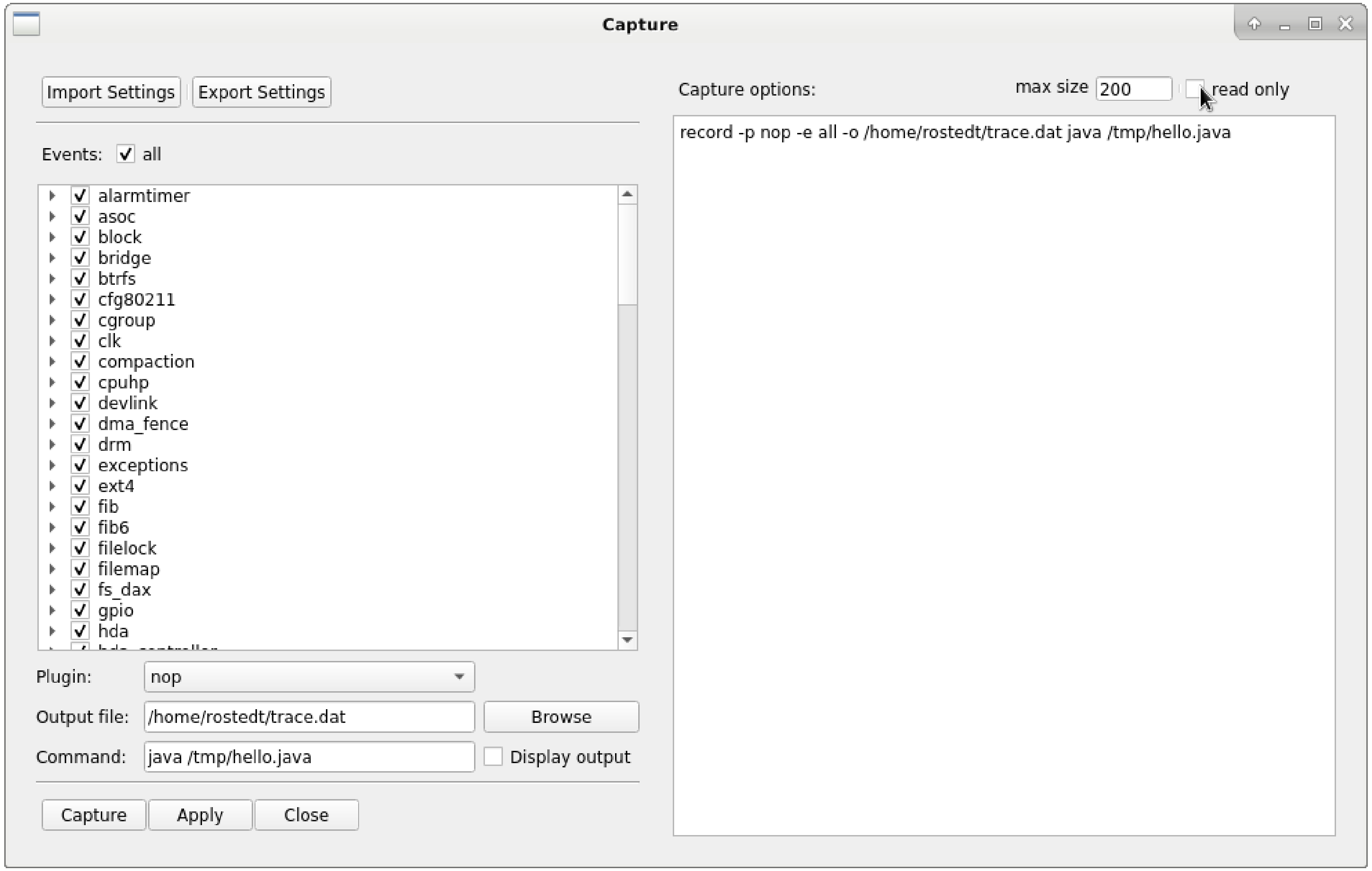
Output file: /home/rostedt/trace.dat  Browse

Command: java /tmp/hello.java  ☐ Display output

Capture | Apply | Close

Output display  max size 200  ✓ read only

# KernelShark 1.0 (Recording)



Capture

Import Settings | Export Settings

Events: ✓ all

- ✓ alarmtimer
- ✓ asoc
- ✓ block
- ✓ bridge
- ✓ btrfs
- ✓ cfg80211
- ✓ cgroup
- ✓ clk
- ✓ compaction
- ✓ cpuhp
- ✓ devlink
- ✓ dma_fence
- ✓ drm
- ✓ exceptions
- ✓ ext4
- ✓ fib
- ✓ fib6
- ✓ filelock
- ✓ filemap
- ✓ fs_dax
- ✓ gpio
- ✓ hda

Plugin: nop

Output file: /home/rostedt/trace.dat

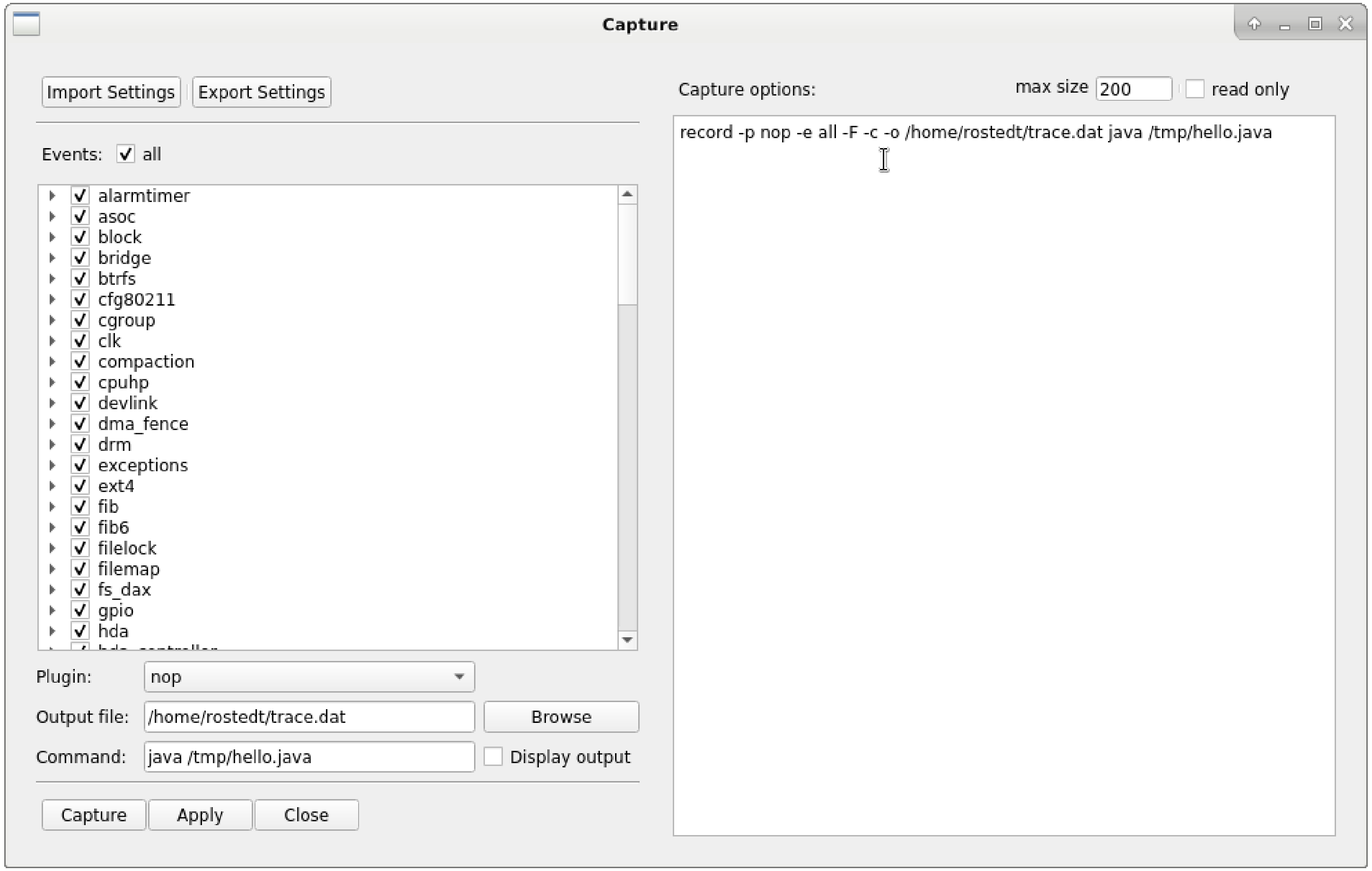
Command: java /tmp/hello.java

Display output: ☐

Capture options: max size 200 ☐ read only

```
record -p nop -e all -o /home/rostedt/trace.dat java /tmp/hello.java
```

# KernelShark 1.0 (Recording)



Capture

Import Settings | Export Settings

Events: ✓ all

- ✓ alarmtimer
- ✓ asoc
- ✓ block
- ✓ bridge
- ✓ btrfs
- ✓ cfg80211
- ✓ cgroup
- ✓ clk
- ✓ compaction
- ✓ cpuhp
- ✓ devlink
- ✓ dma_fence
- ✓ drm
- ✓ exceptions
- ✓ ext4
- ✓ fib
- ✓ fib6
- ✓ filelock
- ✓ filemap
- ✓ fs_dax
- ✓ gpio
- ✓ hda

Plugin: nop

Output file: /home/rostedt/trace.dat

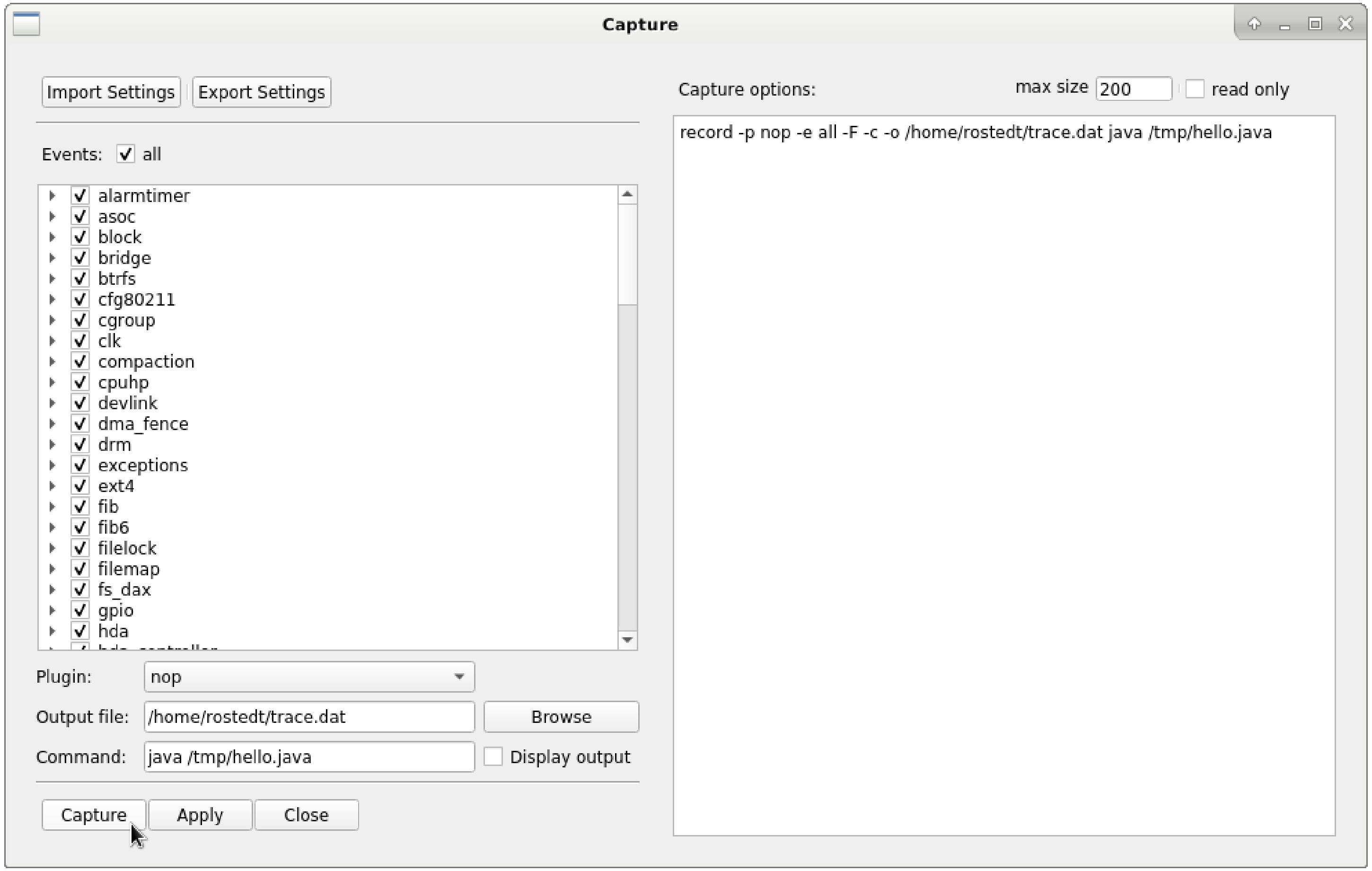
Command: java /tmp/hello.java

Display output: ☐

Capture options: max size 200 ☐ read only

```
record -p nop -e all -F -c -o /home/rostedt/trace.dat java /tmp/hello.java
```

# KernelShark 1.0 (Recording)



Capture options: max size 200 ☐ read only

```
record -p nop -e all -F -c -o /home/rostedt/trace.dat java /tmp/hello.java
```

Events: ✓ all

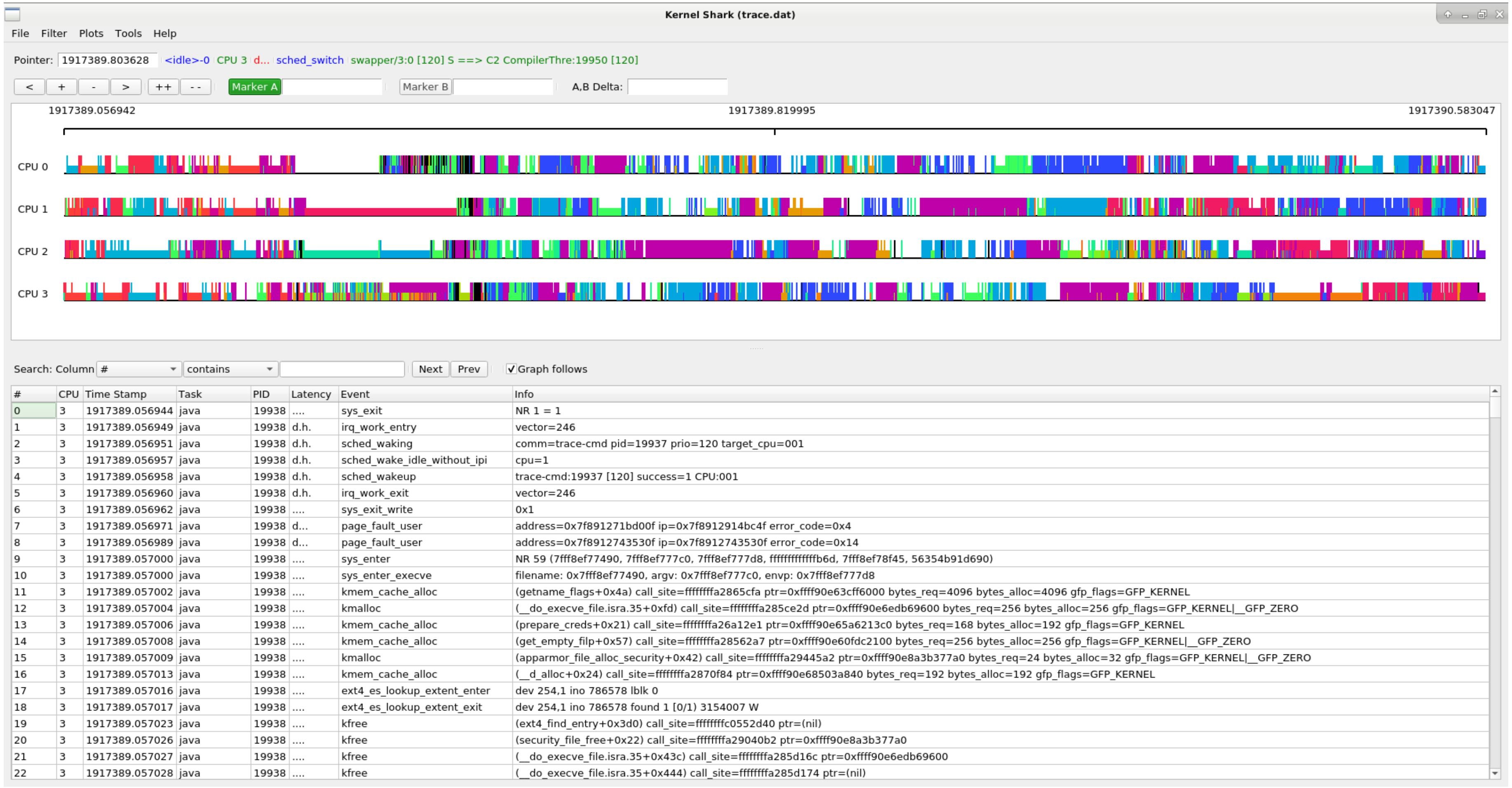
- ✓ alarmtimer
- ✓ asoc
- ✓ block
- ✓ bridge
- ✓ btrfs
- ✓ cfg80211
- ✓ cgroup
- ✓ clk
- ✓ compaction
- ✓ cpuhp
- ✓ devlink
- ✓ dma_fence
- ✓ drm
- ✓ exceptions
- ✓ ext4
- ✓ fib
- ✓ fib6
- ✓ filelock
- ✓ filemap
- ✓ fs_dax
- ✓ gpio
- ✓ hda

Plugin: nop

Output file: /home/rostedt/trace.dat

Command: java /tmp/hello.java ☐ Display output

# KernelShark 1.0 (Simple Java Program)



Pointer: 1917389.803628 <idle>-0 CPU 3 d... sched_switch swapper/3:0 [120] S ==> C2 CompilerThre:19950 [120]

| # | CPU | Time Stamp | Task | PID | Latency | Event | Info |
|---|---|---|---|---|---|---|---|
| 0 | 3 | 1917389.056944 | java | 19938 | .... | sys_exit | NR 1 = 1 |
| 1 | 3 | 1917389.056949 | java | 19938 | d.h. | irq_work_entry | vector=246 |
| 2 | 3 | 1917389.056951 | java | 19938 | d.h. | sched_waking | comm=trace-cmd pid=19937 prio=120 target_cpu=001 |
| 3 | 3 | 1917389.056957 | java | 19938 | d.h. | sched_wake_idle_without_ipi | cpu=1 |
| 4 | 3 | 1917389.056958 | java | 19938 | d.h. | sched_wakeup | trace-cmd:19937 [120] success=1 CPU:001 |
| 5 | 3 | 1917389.056960 | java | 19938 | d.h. | irq_work_exit | vector=246 |
| 6 | 3 | 1917389.056962 | java | 19938 | .... | sys_exit_write | 0x1 |
| 7 | 3 | 1917389.056971 | java | 19938 | d... | page_fault_user | address=0x7f891271bd00f ip=0x7f8912914bc4f error_code=0x4 |
| 8 | 3 | 1917389.056989 | java | 19938 | d... | page_fault_user | address=0x7f8912743530f ip=0x7f8912743530f error_code=0x14 |
| 9 | 3 | 1917389.057000 | java | 19938 | .... | sys_enter | NR 59 (7fff8ef77490, 7fff8ef777c0, 7fff8ef777d8, ffffffffffffb6d, 7fff8ef78f45, 56354b91d690) |
| 10 | 3 | 1917389.057000 | java | 19938 | .... | sys_enter_execve | filename: 0x7fff8ef77490, argv: 0x7fff8ef777c0, envp: 0x7fff8ef777d8 |
| 11 | 3 | 1917389.057002 | java | 19938 | .... | kmem_cache_alloc | (getname_flags+0x4a) call_site=ffffffffa2865cfa ptr=0xffff90e63cff6000 bytes_req=4096 bytes_alloc=4096 gfp_flags=GFP_KERNEL |
| 12 | 3 | 1917389.057004 | java | 19938 | .... | kmalloc | (__do_execve_file.isra.35+0xfd) call_site=ffffffffa285ce2d ptr=0xffff90e6edb69600 bytes_req=256 bytes_alloc=256 gfp_flags=GFP_KERNEL\|__GFP_ZERO |
| 13 | 3 | 1917389.057006 | java | 19938 | .... | kmem_cache_alloc | (prepare_creds+0x21) call_site=ffffffffa26a12e1 ptr=0xffff90e65a6213c0 bytes_req=168 bytes_alloc=192 gfp_flags=GFP_KERNEL |
| 14 | 3 | 1917389.057008 | java | 19938 | .... | kmem_cache_alloc | (get_empty_filp+0x57) call_site=ffffffffa28562a7 ptr=0xffff90e60fdc2100 bytes_req=256 bytes_alloc=256 gfp_flags=GFP_KERNEL\|__GFP_ZERO |
| 15 | 3 | 1917389.057009 | java | 19938 | .... | kmalloc | (apparmor_file_alloc_security+0x42) call_site=ffffffffa29445a2 ptr=0xffff90e8a3b377a0 bytes_req=24 bytes_alloc=32 gfp_flags=GFP_KERNEL\|__GFP_ZERO |
| 16 | 3 | 1917389.057013 | java | 19938 | .... | kmem_cache_alloc | (__d_alloc+0x24) call_site=ffffffffa2870f84 ptr=0xffff90e68503a840 bytes_req=192 bytes_alloc=192 gfp_flags=GFP_KERNEL |
| 17 | 3 | 1917389.057016 | java | 19938 | .... | ext4_es_lookup_extent_enter | dev 254,1 ino 786578 lblk 0 |
| 18 | 3 | 1917389.057017 | java | 19938 | .... | ext4_es_lookup_extent_exit | dev 254,1 ino 786578 found 1 [0/1) 3154007 W |
| 19 | 3 | 1917389.057023 | java | 19938 | .... | kfree | (ext4_find_entry+0x3d0) call_site=ffffffffc0552d40 ptr=(nil) |
| 20 | 3 | 1917389.057026 | java | 19938 | .... | kfree | (security_file_free+0x22) call_site=ffffffffa29040b2 ptr=0xffff90e8a3b377a0 |
| 21 | 3 | 1917389.057027 | java | 19938 | .... | kfree | (__do_execve_file.isra.35+0x43c) call_site=ffffffffa285d16c ptr=0xffff90e6edb69600 |
| 22 | 3 | 1917389.057028 | java | 19938 | .... | kfree | (__do_execve_file.isra.35+0x444) call_site=ffffffffa285d174 ptr=(nil) |

# Thank You