libtracecmd, libtracefs and libtraceevent

Introduction to the Linux kernel tracing libraries

Download examples from:

https://rostedt.org/scale-tracelibs-examples.tar.bz2

or

https://rostedt.org/scale-tracelibs-examples.tar.gz

Introduction to ftrace

- The official tracer of the Linux kernel

Introduction to ftrace

- The official tracer of the Linux kernel
- Added to Linux in 2.6.27 in 2008

Introduction to ftrace

- The official tracer of the Linux kernel
- Added to Linux in 2.6.27 in 2008
- “Ftrace” really is the “function tracer”

Introduction to ftrace

- The official tracer of the Linux kernel
- Added to Linux in 2.6.27 in 2008
- “Ftrace” really is the “function tracer”
    - But also used for the infrastructure that houses the function tracer

Introduction to ftrace

- The official tracer of the Linux kernel
- Added to Linux in 2.6.27 in 2008
- “Ftrace” really is the “function tracer”
  - But also used for the infrastructure that houses the function tracer
- Was designed to be easily used in embedded environments
  - Works with just busybox (cat and echo commands)

Where is ftrace

- If compiled in, there will be a directory: `/sys/kernel/tracing`

Where is ftrace

- If compiled in, there will be a directory: `/sys/kernel/tracing`
- If nothing is there, then mount it:

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

Where is ftrace

- If compiled in, there will be a directory: `/sys/kernel/tracing`
- If nothing is there, then mount it:

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

- No need to do so for using the libraries
  - The libraries will take care of that for you

The tracefs directory

```
# mount -t tracefs tracefs /sys/kernel/tracing
# cd /sys/kernel/tracing
# ls
available_events            hwlat_detector       set_event_notrace_pid    trace_clock
available_filter_functions  instances            set_event_pid            trace_marker
available_tracers           kprobe_events        set_ftrace_filter        trace_marker_raw
buffer_percent              kprobe_profile       set_ftrace_notrace       trace_options
buffer_size_kb              max_graph_depth      set_ftrace_notrace_pid   trace_pipe
buffer_total_size_kb        options              set_ftrace_pid           trace_stat
current_tracer              osnoise              set_graph_function       tracing_cpumask
dynamic_events              per_cpu              set_graph_notrace        tracing_max_latency
dyn_ftrace_total_info       printk_formats       snapshot                 tracing_on
enabled_functions           README               stack_max_size           tracing_thresh
error_log                   recursed_functions   stack_trace              uprobe_events
eval_map                    saved_cmdlines       stack_trace_filter       uprobe_profile
events                      saved_cmdlines_size  synthetic_events         user_events_data
free_buffer                 saved_tgids          timestamp_mode           user_events_status
function_profile_enabled    set_event            trace
```

The tracefs directory

```
# mount -t tracefs tracefs /sys/kernel/tracing
# cd /sys/kernel/tracing
# ls
available_events            hwlat_detector       set_event_notrace_pid   trace_clock
available_filter_functions  instances            set_event_pid           trace_marker
available_tracers           kprobe_events        set_ftrace_filter       trace_marker_raw
buffer_percent              kprobe_profile       set_ftrace_notrace      trace_options
buffer_size_kb              max_graph_depth      set_ftrace_notrace_pid  trace_pipe
buffer_total_size_kb        options              set_ftrace_pid          trace_stat
current_tracer              osnoise              set_graph_function      tracing_cpumask
dynamic_events              per_cpu              set_graph_notrace       tracing_max_latency
dyn_ftrace_total_info       printk_formats       snapshot                tracing_on
enabled_functions           README               stack_max_size          tracing_thresh
error_log                   recursed_functions   stack_trace             uprobe_events
eval_map                    saved_cmdlines       stack_trace_filter      uprobe_profile
events                      saved_cmdlines_size  synthetic_events        user_events_data
free_buffer                 saved_tgids          timestamp_mode          user_events_status
function_profile_enabled    set_event            trace
```

Enabling tracers

```
# echo function > current_tracer
# cat trace
# tracer: function
#
# entries-in-buffer/entries-written: 265996/1500091   #P:8
#
#                                _-----=> irqs-off/BH-disabled
#                               / _----=> need-resched
#                              | / _---=> hardirq/softirq
#                              || / _--=> preempt-depth
#                              ||| / _-=> migrate-disable
#                              |||| /     delay
#           TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#              | |         |   |||||     |         |
          <idle>-0       [006] d..2. 273461.685562: rcu_idle_exit <-cpuidle_enter_state
          <idle>-0       [006] d..3. 273461.685564: rcu_read_lock_sched_held <-trace_cpu_idle
          <idle>-0       [006] d..2. 273461.685564: sched_idle_set_state <-cpuidle_enter_state
          <idle>-0       [006] d..2. 273461.685565: __rcu_irq_enter_check_tick <-rcu_nmi_enter
          <idle>-0       [006] d..3. 273461.685566: rcu_read_lock_sched_held <-trace_hardirqs_off_finish
          <idle>-0       [006] d..2. 273461.685566: irq_enter_rcu <-sysvec_apic_timer_interrupt
          <idle>-0       [006] d..2. 273461.685566: preempt_count_add <-irq_enter_rcu
          <idle>-0       [006] d.h2. 273461.685567: tick_irq_enter <-irq_enter_rcu
          <idle>-0       [006] d.h2. 273461.685567: tick_check_oneshot_broadcast_this_cpu <-tick_irq_enter
          <idle>-0       [006] d.h2. 273461.685567: ktime_get <-tick_irq_enter
```

Enabling tracers

```
# echo function_graph > current_tracer
# cat trace
# tracer: function_graph
#
# CPU  DURATION                  FUNCTION CALLS
# |     |   |                     |   |   |   |
  3)   0.618 us    |                  } /* kvm_steal_clock */
  3)   3.445 us    |                } /* account_process_tick */
  3)               |                run_local_timers() {
  3)   0.675 us    |                  hrtimer_run_queues();
  3)               |                  raise_softirq() {
  3)               |                    __raise_softirq_irqoff() {
  3)               |                      /* softirq_raise: vec=1 [action=TIMER] */
  3)   2.660 us    |                    }
  3)   3.882 us    |                  }
  3)   6.464 us    |                }
  3)               |                rcu_sched_clock_irq() {
  3)               |                  /* rcu_utilization: Start scheduler-tick */
  3)   0.649 us    |                  rcu_is_cpu_rrupt_from_idle();
  3)   0.618 us    |                  rcu_is_cpu_rrupt_from_idle();
  3)   0.614 us    |                  rcu_segcblist_ready_cbs();
  3)               |                  /* rcu_utilization: End scheduler-tick */
  3)   6.336 us    |                }
  3)               |                scheduler_tick() {
  3)   0.801 us    |                  arch_scale_freq_tick();
```

Disabling tracers

```
# echo nop > current_tracer
# cat trace
# tracer: nop
#
# entries-in-buffer/entries-written: 0/0   #P:8
#
#                                _-----=> irqs-off/BH-disabled
#                               / _----=> need-resched
#                              | / _---=> hardirq/softirq
#                              || / _--=> preempt-depth
#                              ||| / _-=> migrate-disable
#                              |||| /     delay
#           TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#              | |         |   |||||     |         |
```

Events

- Broken up into “systems” or “groups”
  - Events are categorized into these systems

Events

- Broken up into “systems” or “groups”
    - Events are categorized into these systems
- Are “static”
    - Exist in the kernel from time the system boots (or module loads) till shutdown

Events

- Broken up into “systems” or “groups”
    - Events are categorized into these systems
- Are “static”
    - Exist in the kernel from time the system boots (or module loads) till shutdown
- Hold data that the developers who created them feel are important

Events

- Broken up into “systems” or “groups”
  - Events are categorized into these systems
- Are “static”
  - Exist in the kernel from time the system boots (or module loads) till shutdown
- Hold data that the developers who created them feel are important
- Can be enabled individually or as a group or all together

Events

- Broken up into “systems” or “groups”
    - Events are categorized into these systems
- Are “static”
    - Exist in the kernel from time the system boots (or module loads) till shutdown
- Hold data that the developers who created them feel are important
- Can be enabled individually or as a group or all together
- Have other features attached to them

Events

- Broken up into “systems” or “groups”
  - Events are categorized into these systems
- Are “static”
  - Exist in the kernel from time the system boots (or module loads) till shutdown
- Hold data that the developers who created them feel are important
- Can be enabled individually or as a group or all together
- Have other features attached to them
  - Triggers - Enable something to happen when the event is hit
  - Histograms - Make a histogram of the data from the event
  - More events - create events based off of the current events

Event systems

```
# ls events/
alarmtimer     dma_fence        huge_memory     jbd2          namei           regmap        thermal
android_fs     drm              hwmon           kmem          napi            rpm           timer
asoc           enable           i2c             kvm           neigh           rseq          tlb
avc            exceptions       i915            kvmmmu        net             rtc           udp
binder         ext4             initcall        kyber         nmi             sched         v4l2
block          fib              intel_iommu     libata        nvme            scsi          vb2
bpf_test_run   fib6             iomap           mac80211      oom             signal        vmscan
bpf_trace      filelock         iommu           mac80211_msg  pagemap         skb           vsock
bridge         filemap          irq             mce           percpu          smbus         vsyscall
cfg80211       fs               irq_matrix      mdio          power           sock          workqueue
cgroup         ftrace           irq_vectors     mei           printk          spi           writeback
clk            gpio             iwlwifi         migrate       qdisc           swiotlb       x86_fpu
compaction     hda              iwlwifi_data    mmap          random          sync_trace    xdp
cpuhp          hda_controller   iwlwifi_io      mmc           ras             syscalls      xhci-hcd
cros_ec        header_event     iwlwifi_msg     module        raw_syscalls    task
devlink        header_page      iwlwifi_ucode   msr           rcu             tcp
```

Event systems

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

Individual events

```
# ls events/sched/sched_switch
enable  filter  format  id  trigger
```

Enabling individual events

```
# echo 1 > events/sched/sched_switch/enable
# cat trace
# tracer: nop
#
# entries-in-buffer/entries-written: 18374/18374   #P:12
#
#                                _-----=> irqs-off
#                               / _----=> need-resched
#                              | / _---=> hardirq/softirq
#                              || / _--=> preempt-depth
#                              ||| /     delay
#           TASK-PID     CPU#  ||||   TIMESTAMP  FUNCTION
#              | |         |   ||||      |         |
          <idle>-0       [000] d..2 67846.361730: sched_switch: prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=R ==> next_comm=kauditd next_pid=75 next_prio=120
         kauditd-75      [000] d..2 67846.361867: sched_switch: prev_comm=kauditd prev_pid=75 prev_prio=120 prev_state=S ==> next_comm=swapper/0 next_pid=0 next_prio=120
          <idle>-0       [002] d..2 67846.361981: sched_switch: prev_comm=swapper/2 prev_pid=0 prev_prio=120 prev_state=R ==> next_comm=auditd next_pid=694 next_prio=116
            sshd-25353   [011] d..2 67846.362032: sched_switch: prev_comm=sshd prev_pid=25353 prev_prio=120 prev_state=D ==> next_comm=swapper/11 next_pid=0 next_prio=120
          <idle>-0       [004] d..2 67846.362085: sched_switch: prev_comm=swapper/4 prev_pid=0 prev_prio=120 prev_state=R ==> next_comm=sslh-fork next_pid=25348 next_prio=120
          <idle>-0       [005] d..2 67846.362094: sched_switch: prev_comm=swapper/5 prev_pid=0 prev_prio=120 prev_state=R ==> next_comm=kworker/u24:1 next_pid=25143 next_prio=120
```

Enabling event systems

```
# echo 1 > events/sched/enable
# cat trace
# tracer: nop
#
# entries-in-buffer/entries-written: 18374/18374   #P:12
#
#                                _-----=> irqs-off
#                               / _----=> need-resched
#                              | / _---=> hardirq/softirq
#                              || / _--=> preempt-depth
#                              ||| /     delay
#           TASK-PID     CPU#  ||||   TIMESTAMP  FUNCTION
#              | |         |   ||||      |         |
            bash-25364   [009] d..3 68831.064923: sched_waking: comm=kworker/u24:3 pid=26148 prio=120 target_cpu=005
            bash-25364   [009] d..4 68831.064952: sched_wakeup: comm=kworker/u24:3 pid=26148 prio=120 target_cpu=005
            bash-25364   [009] d.h1 68831.064987: sched_stat_runtime: comm=bash pid=25364 runtime=913246 [ns] vruntime=294863255944 [ns]
            bash-25364   [009] d.s3 68831.065003: sched_waking: comm=rcu_preempt pid=14 prio=120 target_cpu=007
            bash-25364   [009] d.s4 68831.065018: sched_wakeup: comm=rcu_preempt pid=14 prio=120 target_cpu=007
            bash-25364   [009] d..2 68831.065091: sched_stat_runtime: comm=bash pid=25364 runtime=103491 [ns] vruntime=294863359435 [ns]
            bash-25364   [009] d..2 68831.065097: sched_switch: prev_comm=bash prev_pid=25364 prev_prio=120 prev_state=S ==> next_comm=swapper/9 next_pid=0 next_prio=120
          <idle>-0       [005] d..2 68831.065185: sched_switch: prev_comm=swapper/5 prev_pid=0 prev_prio=120 prev_state=R ==> next_comm=kworker/u24:3 next_pid=26148 next_prio=120
          <idle>-0       [007] d..2 68831.065185: sched_switch: prev_comm=swapper/7 prev_pid=0 prev_prio=120 prev_state=R ==> next_comm=rcu_preempt next_pid=14 next_prio=120
     rcu_preempt-14      [007] d..2 68831.065204: sched_stat_runtime: comm=rcu_preempt pid=14 runtime=195450 [ns] vruntime=237201695252 [ns]
   kworker/u24:3-26148   [005] d..3 68831.065210: sched_waking: comm=sshd pid=25353 prio=120 target_cpu=004
     rcu_preempt-14      [007] d..2 68831.065210: sched_switch: prev_comm=rcu_preempt prev_pid=14 prev_prio=120 prev_state=I ==> next_comm=swapper/7 next_pid=0 next_prio=120
```

Enabling all events

```
# echo 1 > events/enable
# cat trace
# tracer: nop
#
# entries-in-buffer/entries-written: 18374/18374   #P:12
#
#                                _-----=> irqs-off
#                               / _----=> need-resched
#                              | / _---=> hardirq/softirq
#                              || / _--=> preempt-depth
#                              ||| /     delay
#           TASK-PID     CPU#  ||||   TIMESTAMP  FUNCTION
#              | |         |   ||||      |         |
            bash-9183    [006] .... 171201.802642: writeback_mark_inode_dirty: bdi (unknown): ino=10589 state=I_DIRTY_SYNC|I_DIRTY_DATASYNC|I_DIRTY_PAGES flags=I_DIRTY_SYNC|I_DIRTY_DAT
            bash-9183    [006] .... 171201.802647: writeback_dirty_inode_start: bdi (unknown): ino=10589 state=I_DIRTY_SYNC|I_DIRTY_DATASYNC|I_DIRTY_PAGES flags=I_DIRTY_SYNC|I_DIRTY_DA
            bash-9183    [006] .... 171201.802648: writeback_dirty_inode: bdi (unknown): ino=10589 state=I_DIRTY_SYNC|I_DIRTY_DATASYNC|I_DIRTY_PAGES flags=I_DIRTY_SYNC|I_DIRTY_DATASYNC
            bash-9183    [006] .... 171201.802662: do_sys_open: "trace" 8241 666
            bash-9183    [006] .... 171201.802667: kmem_cache_free: call_site=do_sys_openat2+0x17b/0x1ab ptr=00000000b7017b15
            bash-9183    [006] .... 171201.802672: kfree: call_site=__audit_syscall_exit+0x1b9/0x23f ptr=0000000000000000
            bash-9183    [006] ...1 171201.802673: sys_openat -> 0x3
            bash-9183    [006] .... 171201.802675: sys_exit: NR 257 = 3
            bash-9183    [006] ...1 171201.802716: sys_fcntl(fd: 1, cmd: 1, arg: 0)
            bash-9183    [006] .... 171201.802717: sys_enter: NR 72 (1, 1, 0, 1b6, 7, 5c9ce7813dd0)
            bash-9183    [006] .... 171201.802723: kfree: call_site=__audit_syscall_exit+0x1b9/0x23f ptr=0000000000000000
            bash-9183    [006] ...1 171201.802724: sys_fcntl -> 0x0
            bash-9183    [006] .... 171201.802725: sys_exit: NR 72 = 0
            bash-9183    [006] ...1 171201.802726: sys_fcntl(fd: 1, cmd: 0, arg: a)
            bash-9183    [006] .... 171201.802726: sys_enter: NR 72 (1, 0, a, 1b6, 7, 5c9ce7813dd0)
```

Clearing the trace file

```
# echo 0 > tracing_on
# echo > trace
# cat trace
# tracer: nop
#
# entries-in-buffer/entries-written: 0/0   #P:12
#
#                                _-----=> irqs-off
#                               / _----=> need-resched
#                              | / _---=> hardirq/softirq
#                              || / _--=> preempt-depth
#                              ||| /     delay
#           TASK-PID     CPU#  ||||   TIMESTAMP  FUNCTION
#              | |         |   ||||      |         |
```

Reading the trace

- The human readable files
  - trace
    - Reads the trace in a non destructive mode
      - With tracing off, will produce the same output, each time it is read
    - Can produce side-effects when reading while tracing is happening

Reading the trace

- The human readable files
  - trace
    - Reads the trace in a non destructive mode
      - With tracing off, will produce the same output, each time it is read
    - Can produce side-effects when reading while tracing is happening
  - trace_pipe
    - Reads the trace in a produce / consumer mode
      - Will consume the trace
      - Will not produce the same output each time it is read
      - With tracing off, can empty the trace buffer
    - No side-effects when reading while tracing is happening

Per CPU trace files

- Located in `/sys/kernel/tracing/per_cpu/cpuX`
  - Where X is the CPU number

Per CPU trace files

- Located in `/sys/kernel/tracing/per_cpu/cpuX`
  - Where X is the CPU number
- trace
  - Same as the top level `trace` file but only shows the trace data for the given CPU
- trace_pipe
  - Same as the top level `trace_pipe` file but only shows the trace data for the given CPU
- trace_pipe_raw
  - This extracts the raw trace data from the ring buffer (binary format, not ASCII)

Per CPU trace files

- Located in `/sys/kernel/tracing/per_cpu/cpuX`
  - Where X is the CPU number
- trace
  - Same as the top level `trace` file but only shows the trace data for the given CPU
- trace_pipe
  - Same as the top level `trace_pipe` file but only shows the trace data for the given CPU
- trace_pipe_raw
  - This extracts the raw trace data from the ring buffer (binary format, not ASCII)
  - Read in “sub-buffers” defined by `/sys/kernel/tracing/events/header_page`
    - The events defined in `/sys/kernel/tracing/events/header_event`

Per CPU raw data files

```
# cd /sys/kernel/tracing
# cat events/header_page
      field: u64 timestamp;   offset:0;   size:8;     signed:0;
      field: local_t commit;  offset:8;   size:8;     signed:1;
      field: int overwrite;   offset:8;   size:1;     signed:1;
      field: char data; offset:16;  size:4080;  signed:1;
```

Per CPU raw data files

```
# cd /sys/kernel/tracing
# cat events/header_page
      field: u64 timestamp;   offset:0;   size:8;     signed:0;
      field: local_t commit;  offset:8;   size:8;     signed:1;
      field: int overwrite;   offset:8;   size:1;     signed:1;
      field: char data; offset:16;  size:4080;  signed:1;

# echo 1 > events/sched/sched_switch/enable
# cat per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
```

Per CPU raw data files

```
# cd /sys/kernel/tracing
# cat events/header_page
      field: u64 timestamp;   offset:0;   size:8;     signed:0;
      field: local_t commit;  offset:8;   size:8;     signed:1;
      field: int overwrite;   offset:8;   size:1;     signed:1;
      field: char data; offset:16;  size:4080;  signed:1;

# echo 1 > events/sched/sched_switch/enable
# cat per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C
```

Per CPU raw data files

```
# cd /sys/kernel/tracing
# cat events/header_page
      field: u64 timestamp;   offset:0;   size:8;     signed:0;
      field: local_t commit;  offset:8;   size:8;     signed:1;
      field: int overwrite;   offset:8;   size:1;     signed:1;
      field: char data; offset:16;  size:4080;  signed:1;

# echo 1 > events/sched/sched_switch/enable
# cat per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# hexdump -C /tmp/raw0 | head
00000000  f2 94 d0 da d0 07 00 00  7c 0c 00 00 00 00 00 00  |........|.......|
00000010  10 44 a0 00 34 01 01 02  53 09 00 00 63 61 74 00  |.D..4...S...cat.|
00000020  00 00 00 00 00 00 00 00  00 00 00 00 53 09 00 00  |............S...|
00000030  78 00 00 00 20 00 00 00  00 00 00 00 73 77 61 70  |x... .......swap|
00000040  70 65 72 2f 30 00 00 00  00 00 00 00 00 00 00 00  |per/0...........|
00000050  78 00 00 00 9e ce 27 fe  01 00 00 00 10 00 00 00  |x.....'.........|
00000060  34 01 01 02 00 00 00 00  73 77 61 70 70 65 72 2f  |4.......swapper/|
00000070  30 00 00 00 00 00 00 00  00 00 00 00 78 00 00 00  |0...........x...|
00000080  00 00 00 00 00 00 00 00  6b 77 6f 72 6b 65 72 2f  |........kworker/|
00000090  30 3a 31 00 00 00 00 00  f3 07 00 00 78 00 00 00  |0:1.........x...|
```

Per CPU raw data files

```
# cd /sys/kernel/tracing
# cat events/header_page
      field: u64 timestamp;   offset:0;   size:8;     signed:0;
      field: local_t commit;  offset:8;   size:8;     signed:1;
      field: int overwrite;   offset:8;   size:1;     signed:1;
      field: char data; offset:16;  size:4080;  signed:1;

# echo 1 > events/sched/sched_switch/enable
# cat per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# hexdump -C /tmp/raw0 | head
00000000  f2 94 d0 da d0 07 00 00  7c 0c 00 00 00 00 00 00  |........|.......|
00000010  10 44 a0 00 34 01 01 02  53 09 00 00 63 61 74 00  |.D..4...S...cat.|
00000020  00 00 00 00 00 00 00 00  00 00 00 00 53 09 00 00  |............S...|
00000030  78 00 00 00 20 00 00 00  00 00 00 00 73 77 61 70  |x... .......swap|
00000040  70 65 72 2f 30 00 00 00  00 00 00 00 00 00 00 00  |per/0...........|
00000050  78 00 00 00 9e ce 27 fe  01 00 00 00 10 00 00 00  |x.....'.........|
00000060  34 01 01 02 00 00 00 00  73 77 61 70 70 65 72 2f  |4.......swapper/|
00000070  30 00 00 00 00 00 00 00  00 00 00 00 78 00 00 00  |0...........x...|
00000080  00 00 00 00 00 00 00 00  6b 77 6f 72 6b 65 72 2f  |........kworker/|
00000090  30 3a 31 00 00 00 00 00  f3 07 00 00 78 00 00 00  |0:1.........x...|
```

Per CPU raw data files

```
# cd /sys/kernel/tracing
# cat events/header_page
      field: u64 timestamp;   offset:0;   size:8;     signed:0;
      field: local_t commit;  offset:8;   size:8;     signed:1;
      field: int overwrite;   offset:8;   size:1;     signed:1;
      field: char data; offset:16;  size:4080;  signed:1;

# echo 1 > events/sched/sched_switch/enable
# cat per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# hexdump -C /tmp/raw0 | head
00000000  f2 94 d0 da d0 07 00 00  7c 0c 00 00 00 00 00 00  |........|.......|
00000010  10 44 a0 00 34 01 01 02  53 09 00 00 63 61 74 00  |.D..4...S...cat.|
00000020  00 00 00 00 00 00 00 00  00 00 00 00 53 09 00 00  |............S...|
00000030  78 00 00 00 20 00 00 00  00 00 00 00 73 77 61 70  |x... .......swap|
00000040  70 65 72 2f 30 00 00 00  00 00 00 00 00 00 00 00  |per/0...........|
00000050  78 00 00 00 9e ce 27 fe  01 00 00 00 10 00 00 00  |x.....'.........|
00000060  34 01 01 02 00 00 00 00  73 77 61 70 70 65 72 2f  |4.......swapper/|
00000070  30 00 00 00 00 00 00 00  00 00 00 00 78 00 00 00  |0...........x...|
00000080  00 00 00 00 00 00 00 00  6b 77 6f 72 6b 65 72 2f  |........kworker/|
00000090  30 3a 31 00 00 00 00 00  f3 07 00 00 78 00 00 00  |0:1.........x...|
```

Per CPU raw data files

```
# cd /sys/kernel/tracing
# cat events/header_page
      field: u64 timestamp;   offset:0;   size:8;     signed:0;
      field: local_t commit;  offset:8;   size:8;     signed:1;
      field: int overwrite;   offset:8;   size:1;     signed:1;
      field: char data; offset:16;  size:4080;  signed:1;

# echo 1 > events/sched/sched_switch/enable
# cat per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# hexdump -C /tmp/raw0 | head
00000000  f2 94 d0 da d0 07 00 00  7c 0c 00 00 00 00 00 00  |........|.......|
00000010  10 44 a0 00 34 01 01 02  53 09 00 00 63 61 74 00  |.D..4...S...cat.|
00000020  00 00 00 00 00 00 00 00  00 00 00 00 53 09 00 00  |............S...|
00000030  78 00 00 00 20 00 00 00  00 00 00 00 73 77 61 70  |x... .......swap|
00000040  70 65 72 2f 30 00 00 00  00 00 00 00 00 00 00 00  |per/0...........|
00000050  78 00 00 00 9e ce 27 fe  01 00 00 00 10 00 00 00  |x.....'.........|
00000060  34 01 01 02 00 00 00 00  73 77 61 70 70 65 72 2f  |4.......swapper/|
00000070  30 00 00 00 00 00 00 00  00 00 00 00 78 00 00 00  |0...........x...|
00000080  00 00 00 00 00 00 00 00  6b 77 6f 72 6b 65 72 2f  |........kworker/|
00000090  30 3a 31 00 00 00 00 00  f3 07 00 00 78 00 00 00  |0:1.........x...|
```

Per CPU raw data files

```
# cd /sys/kernel/tracing
# cat events/header_page
      field: u64 timestamp;   offset:0;   size:8;     signed:0;
      field: local_t commit;  offset:8;   size:8;     signed:1;
      field: int overwrite;   offset:8;   size:1;     signed:1;
      field: char data; offset:16;  size:4080;  signed:1;

# echo 1 > events/sched/sched_switch/enable
# cat per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# hexdump -C /tmp/raw0 | head
00000000  f2 94 d0 da d0 07 00 00  7c 0c 00 00 00 00 00 00  |........|.......|
00000010  10 44 a0 00 34 01 01 02  53 09 00 00 63 61 74 00  |.D..4...S...cat.|
00000020  00 00 00 00 00 00 00 00  00 00 00 00 53 09 00 00  |............S...|
00000030  78 00 00 00 20 00 00 00  00 00 00 00 73 77 61 70  |x... .......swap|
00000040  70 65 72 2f 30 00 00 00  00 00 00 00 00 00 00 00  |per/0...........|
00000050  78 00 00 00 9e ce 27 fe  01 00 00 00 10 00 00 00  |x.....'.........|
00000060  34 01 01 02 00 00 00 00  73 77 61 70 70 65 72 2f  |4.......swapper/|
00000070  30 00 00 00 00 00 00 00  00 00 00 00 78 00 00 00  |0...........x...|
00000080  00 00 00 00 00 00 00 00  6b 77 6f 72 6b 65 72 2f  |........kworker/|
00000090  30 3a 31 00 00 00 00 00  f3 07 00 00 78 00 00 00  |0:1.........x...|
```

Per CPU raw data files

```
# cd /sys/kernel/tracing
# cat events/header_event
# compressed entry header
      type_len    :    5 bits
      time_delta  :   27 bits
      array       :   32 bits

      padding     : type == 29
      time_extend : type == 30
      time_stamp : type == 31
      data max type_len  == 28
```

Per CPU raw data files

```
# cd /sys/kernel/tracing
# cat events/header_event
# compressed entry header
	type_len    :    5 bits
	time_delta  :   27 bits
	array       :   32 bits

	padding     : type == 29
	time_extend : type == 30
	time_stamp : type == 31
	data max type_len  == 28

# hexdump -C /tmp/raw0 | head
00000000  f2 94 d0 da d0 07 00 00  7c 0c 00 00 00 00 00 00  |........|.......|
00000010  10 44 a0 00 34 01 01 02  53 09 00 00 63 61 74 00  |.D..4...S...cat.|
00000020  00 00 00 00 00 00 00 00  00 00 00 00 53 09 00 00  |............S...|
00000030  78 00 00 00 20 00 00 00  00 00 00 00 73 77 61 70  |x... .......swap|
00000040  70 65 72 2f 30 00 00 00  00 00 00 00 00 00 00 00  |per/0...........|
00000050  78 00 00 00 9e ce 27 fe  01 00 00 00 10 00 00 00  |x.....'.........|
00000060  34 01 01 02 00 00 00 00  73 77 61 70 70 65 72 2f  |4.......swapper/|
00000070  30 00 00 00 00 00 00 00  00 00 00 00 78 00 00 00  |0...........x...|
00000080  00 00 00 00 00 00 00 00  6b 77 6f 72 6b 65 72 2f  |........kworker/|
00000090  30 3a 31 00 00 00 00 00  f3 07 00 00 78 00 00 00  |0:1.........x...|
```

Per CPU raw data files

```
# cd /sys/kernel/tracing
# cat events/header_event
# compressed entry header
      type_len    :    5 bits
      time_delta  :   27 bits
      array       :   32 bits

      padding     : type == 29
      time_extend : type == 30
      time_stamp : type == 31
      data max type_len  == 28

# hexdump -C /tmp/raw0 | head
00000000  f2 94 d0 da d0 07 00 00  7c 0c 00 00 00 00 00 00  |........|.......|
00000010  10 44 a0 00 34 01 01 02  53 09 00 00 63 61 74 00  |.D..4...S...cat.|
00000020  00 00 00 00 00 00 00 00  00 00 00 00 53 09 00 00  |............S...|
00000030  78 00 00 00 20 00 00 00  00 00 00 00 73 77 61 70  |x... .......swap|
00000040  70 65 72 2f 30 00 00 00  00 00 00 00 00 00 00 00  |per/0...........|
00000050  78 00 00 00 9e ce 27 fe  01 00 00 00 10 00 00 00  |x.....'.........|
00000060  34 01 01 02 00 00 00 00  73 77 61 70 70 65 72 2f  |4.......swapper/|
00000070  30 00 00 00 00 00 00 00  00 00 00 00 78 00 00 00  |0...........x...|
00000080  00 00 00 00 00 00 00 00  6b 77 6f 72 6b 65 72 2f  |........kworker/|
00000090  30 3a 31 00 00 00 00 00  f3 07 00 00 78 00 00 00  |0:1.........x...|
```

Per CPU raw data files

```
# cd /sys/kernel/tracing
# cat events/header_event
# compressed entry header
	type_len    :    5 bits
	time_delta  :   27 bits
	array       :   32 bits

	padding     : type == 29
	time_extend : type == 30
	time_stamp : type == 31
	data max type_len  == 28

# hexdump -C /tmp/raw0 | head
00000000  f2 94 d0 da d0 07 00 00  7c 0c 00 00 00 00 00 00  |........|.......|
00000010  10 44 a0 00 34 01 01 02  53 09 00 00 63 61 74 00  |.D..4...S...cat.|
00000020  00 00 00 00 00 00 00 00  00 00 00 00 53 09 00 00  |............S...|
00000030  78 00 00 00 20 00 00 00  00 00 00 00 73 77 61 70  |x... .......swap|
00000040  70 65 72 2f 30 00 00 00  00 00 00 00 00 00 00 00  |per/0...........|
00000050  78 00 00 00 9e ce 27 fe  01 00 00 00 10 00 00 00  |x.....'.........|
00000060  34 01 01 02 00 00 00 00  73 77 61 70 70 65 72 2f  |4.......swapper/|
00000070  30 00 00 00 00 00 00 00  00 00 00 00 78 00 00 00  |0...........x...|
00000080  00 00 00 00 00 00 00 00  6b 77 6f 72 6b 65 72 2f  |........kworker/|
00000090  30 3a 31 00 00 00 00 00  f3 07 00 00 78 00 00 00  |0:1.........x...|
```

Event formats

```
# cat events/sched/sched_switch/format
name: sched_switch
ID: 334
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), "|", { 0x0001, "S" }, { 0x0002, "D" }, { 0x0004, "T" }, { 0x0008, "t" }, {
0x0010, "X" }, { 0x0020, "Z" }, { 0x0040, "P" }, { 0x0080, "I" }) : "R", REC->prev_state & (((0x0000
| 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010 | 0x0020 | 0x0040) + 1) << 1) ? "+" : "",
REC->next_comm, REC->next_pid, REC->next_prio
```

Event formats

```
# cat events/sched/sched_switch/format
name: sched_switch
ID: 334
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), "|", { 0x0001, "S" }, { 0x0002, "D" }, { 0x0004, "T" }, { 0x0008, "t" }, {
0x0010, "X" }, { 0x0020, "Z" }, { 0x0040, "P" }, { 0x0080, "I" }) : "R", REC->prev_state & (((0x0000
| 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010 | 0x0020 | 0x0040) + 1) << 1) ? "+" : "",
REC->next_comm, REC->next_pid, REC->next_prio
```

Event formats

```
# cat events/sched/sched_switch/format
name: sched_switch
ID: 334
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), "|", { 0x0001, "S" }, { 0x0002, "D" }, { 0x0004, "T" }, { 0x0008, "t" }, {
0x0010, "X" }, { 0x0020, "Z" }, { 0x0040, "P" }, { 0x0080, "I" }) : "R", REC->prev_state & (((0x0000
| 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010 | 0x0020 | 0x0040) + 1) << 1) ? "+" : "",
REC->next_comm, REC->next_pid, REC->next_prio
```

Event formats

```
# cat events/sched/sched_switch/format
name: sched_switch
ID: 334
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), "|", { 0x0001, "S" }, { 0x0002, "D" }, { 0x0004, "T" }, { 0x0008, "t" }, {
0x0010, "X" }, { 0x0020, "Z" }, { 0x0040, "P" }, { 0x0080, "I" }) : "R", REC->prev_state & (((0x0000
| 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010 | 0x0020 | 0x0040) + 1) << 1) ? "+" : "",
REC->next_comm, REC->next_pid, REC->next_prio
```

The Libraries

- libtraceevent
  - Functions to parse the trace event formats to read the raw events
- libtracefs
  - Functions to access the tracefs file system (`/sys/kernel/tracing`)
- libtracecmd
  - Functions to open and create the `trace.dat` file produced by trace-cmd

Building the libraries

May be already provided by the distributions (but best to use the latest)

```
$ git clone git://git.kernel.org/pub/scm/libs/libtrace/libtraceevent.git
$ cd libtraceevent
$ make
$ make doc
$ sudo make install
$ sudo make doc-install
```

Building the libraries

May be already provided by the distributions (but best to use the latest)

```
$ git clone git://git.kernel.org/pub/scm/libs/libtrace/libtraceevent.git
$ cd libtraceevent
$ make
$ make doc
$ sudo make install
$ sudo make doc-install

$ git clone git://git.kernel.org/pub/scm/libs/libtrace/libtracefs.git
$ cd libtracefs
$ make
$ make doc
$ sudo make install
$ sudo make install_doc
```

Building the libraries

May be already provided by the distributions (but best to use the latest)

```
$ git clone git://git.kernel.org/pub/scm/libs/libtrace/libtraceevent.git
$ cd libtraceevent
$ make
$ make doc
$ sudo make install
$ sudo make doc-install

$ git clone git://git.kernel.org/pub/scm/libs/libtrace/libtracefs.git
$ cd libtracefs
$ make
$ make doc
$ sudo make install
$ sudo make install_doc

$ git clone git://git.kernel.org/pub/scm/utils/trace-cmd/trace-cmd.git
$ cd trace-cmd
$ make libs
$ make doc
$ sudo make install libs
$ sudo make install_doc
```

libtraceevent

- Reads the format files to know how to parse the raw events

libtraceevent

- Reads the format files to know how to parse the raw events
- First of the libraries to be created

libtraceevent

- Reads the format files to know how to parse the raw events
- First of the libraries to be created
    - Created for trace-cmd

libtraceevent

- Reads the format files to know how to parse the raw events
- First of the libraries to be created
  - Created for trace-cmd
  - Ported over to perf (maintained in two different locations)

libtraceevent

- Reads the format files to know how to parse the raw events
- First of the libraries to be created
    - Created for trace-cmd
    - Ported over to perf (maintained in two different locations)
    - Ported to powertop (maintained in a third location)

libtraceevent

- Reads the format files to know how to parse the raw events
- First of the libraries to be created
  - Created for trace-cmd
  - Ported over to perf (maintained in two different locations)
  - Ported to powertop (maintained in a third location)
  - Poorly designed (we all learn how on our first try)

libtraceevent

- Reads the format files to know how to parse the raw events
- First of the libraries to be created
  - Created for trace-cmd
  - Ported over to perf (maintained in two different locations)
  - Ported to powertop (maintained in a third location)
  - Poorly designed (we all learn how on our first try)
- Header files are:
  - event-parse.h
  - kbuffer.h
  - trace-seq.h

Simple tep example (simple-tep.c)

```c
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
#include <event-parse.h>
#include <kbuffer.h>
#include <trace-seq.h>

static char *read_file(const char *file)
{
      char buf[BUFSIZ];
      char *f = NULL;
      int fd, r, size = 0;

      fd = open(file, O_RDONLY);
      do {
            r = read(fd, buf, BUFSIZ);
            f = realloc(f, r + size); // unsafe!
            memcpy(f + size, buf, r);
            size += r;
      } while (r > 0);

      return f;
}
```

Simple tep example (simple-tep.c)

```
#include <stdlib.h>
#include <fcntl.h>
#include <unistd.h>
#include <event-parse.h>
#include <kbuffer.h>
#include <trace-seq.h>

static char *read_file(const char *file)
{
      char buf[BUFSIZ];
      char *f = NULL;
      int fd, r, size = 0;

      fd = open(file, O_RDONLY);
      do {
            r = read(fd, buf, BUFSIZ);
            f = realloc(f, r + size); // unsafe!
            memcpy(f + size, buf, r);
            size += r;
      } while (r > 0);

      return f;
}
```

Simple tep example (simple-tep.c)

```
int main(int argc, char **argv) {
      struct tep_record record;
      struct tep_handle *tep;
      struct kbuffer *kbuf;
      struct trace_seq seq;
      unsigned long long ts;
      void *buf;
      int sub_buf_size;
      int fd;

      fd = open(argv[1], O_RDONLY);

      trace_seq_init(&seq);
      tep = tep_alloc();
      buf = read_file("/sys/kernel/tracing/events/header_page");
      tep_parse_header_page(tep, buf, strlen(buf), 0);
      free(buf);
      buf = read_file("/sys/kernel/tracing/events/sched/sched_switch/format");
      tep_parse_event(tep, buf, strlen(buf), "sched");
      free(buf);
```

Simple tep example (simple-tep.c)

```c
int main(int argc, char **argv) {
      struct tep_record record;
      struct tep_handle *tep;
      struct kbuffer *kbuf;
      struct trace_seq seq;
      unsigned long long ts;
      void *buf;
      int sub_buf_size;
      int fd;

      fd = open(argv[1], O_RDONLY);

      trace_seq_init(&seq);
      tep = tep_alloc();
      buf = read_file("/sys/kernel/tracing/events/header_page");
      tep_parse_header_page(tep, buf, strlen(buf), 0);
      free(buf);
      buf = read_file("/sys/kernel/tracing/events/sched/sched_switch/format");
      tep_parse_event(tep, buf, strlen(buf), "sched");
      free(buf);
```

Simple tep example (simple-tep.c)

```
int main(int argc, char **argv) {
      struct tep_record record;
      struct tep_handle *tep;
      struct kbuffer *kbuf;
      struct trace_seq seq;
      unsigned long long ts;
      void *buf;
      int sub_buf_size;
      int fd;

      fd = open(argv[1], O_RDONLY);

      trace_seq_init(&seq);
      tep = tep_alloc();
      buf = read_file("/sys/kernel/tracing/events/header_page");
      tep_parse_header_page(tep, buf, strlen(buf), 0);
      free(buf);
      buf = read_file("/sys/kernel/tracing/events/sched/sched_switch/format");
      tep_parse_event(tep, buf, strlen(buf), "sched");
      free(buf);
```

Simple tep example (simple-tep.c)

```
int main(int argc, char **argv) {
      struct tep_record record;
      struct tep_handle *tep;
      struct kbuffer *kbuf;
      struct trace_seq seq;
      unsigned long long ts;
      void *buf;
      int sub_buf_size;
      int fd;

      fd = open(argv[1], O_RDONLY);

      trace_seq_init(&seq);
      tep = tep_alloc();
      buf = read_file("/sys/kernel/tracing/events/header_page");
      tep_parse_header_page(tep, buf, strlen(buf), 0);
      free(buf);
      buf = read_file("/sys/kernel/tracing/events/sched/sched_switch/format");
      tep_parse_event(tep, buf, strlen(buf), "sched");
      free(buf);
```

Simple tep example (simple-tep.c)

```
int main(int argc, char **argv) {
      struct tep_record record;
      struct tep_handle *tep;
      struct kbuffer *kbuf;
      struct trace_seq seq;
      unsigned long long ts;
      void *buf;
      int sub_buf_size;
      int fd;

      fd = open(argv[1], O_RDONLY);

      trace_seq_init(&seq);
      tep = tep_alloc();
      buf = read_file("/sys/kernel/tracing/events/header_page");
      tep_parse_header_page(tep, buf, strlen(buf), 0);
      free(buf);
      buf = read_file("/sys/kernel/tracing/events/sched/sched_switch/format");
      tep_parse_event(tep, buf, strlen(buf), "sched");
      free(buf);
```

Simple tep example (simple-tep.c)

```
      sub_buf_size = tep_get_sub_buffer_size(tep);
      buf = malloc(sub_buf_size);
      read(fd, buf, sub_buf_size);
      kbuf = kbuffer_alloc(tep_get_header_page_size(tep) == 8, !tep_is_bigendian());
      kbuffer_load_subbuffer(kbuf, buf);
      record.data = kbuffer_read_event(kbuf, &ts);
      record.ts = ts;
      record.missed_events = kbuffer_missed_events(kbuf);
      record.size = kbuffer_event_size(kbuf);
      record.record_size = kbuffer_curr_size(kbuf);
      record.cpu = 0;
      kbuffer_next_event(kbuf, NULL);
      tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
                  TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                  TEP_PRINT_NAME, TEP_PRINT_INFO);
      trace_seq_do_printf(&seq);
      trace_seq_destroy(&seq);

      return 0;
}
```

Simple tep example (simple-tep.c)

```
	sub_buf_size = tep_get_sub_buffer_size(tep);
	buf = malloc(sub_buf_size);
	read(fd, buf, sub_buf_size);
	kbuf = kbuffer_alloc(tep_get_header_page_size(tep) == 8, !tep_is_bigendian());
	kbuffer_load_subbuffer(kbuf, buf);
	record.data = kbuffer_read_event(kbuf, &ts);
	record.ts = ts;
	record.missed_events = kbuffer_missed_events(kbuf);
	record.size = kbuffer_event_size(kbuf);
	record.record_size = kbuffer_curr_size(kbuf);
	record.cpu = 0;
	kbuffer_next_event(kbuf, NULL);
	tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
			TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
			TEP_PRINT_NAME, TEP_PRINT_INFO);
	trace_seq_do_printf(&seq);
	trace_seq_destroy(&seq);

	return 0;
}
```

Simple tep example (simple-tep.c)

```
        sub_buf_size = tep_get_sub_buffer_size(tep);
        buf = malloc(sub_buf_size);
        read(fd, buf, sub_buf_size);
        kbuf = kbuffer_alloc(tep_get_header_page_size(tep) == 8, !tep_is_bigendian());
        kbuffer_load_subbuffer(kbuf, buf);
        record.data = kbuffer_read_event(kbuf, &ts);
        record.ts = ts;
        record.missed_events = kbuffer_missed_events(kbuf);
        record.size = kbuffer_event_size(kbuf);
        record.record_size = kbuffer_curr_size(kbuf);
        record.cpu = 0;
        kbuffer_next_event(kbuf, NULL);
        tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
                        TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                        TEP_PRINT_NAME, TEP_PRINT_INFO);
        trace_seq_do_printf(&seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Simple tep example (simple-tep.c)

```
        sub_buf_size = tep_get_sub_buffer_size(tep);
        buf = malloc(sub_buf_size);
        read(fd, buf, sub_buf_size);
        kbuf = kbuffer_alloc(tep_get_header_page_size(tep) == 8, !tep_is_bigendian());
        kbuffer_load_subbuffer(kbuf, buf);
        record.data = kbuffer_read_event(kbuf, &ts);
        record.ts = ts;
        record.missed_events = kbuffer_missed_events(kbuf);
        record.size = kbuffer_event_size(kbuf);
        record.record_size = kbuffer_curr_size(kbuf);
        record.cpu = 0;
        kbuffer_next_event(kbuf, NULL);
        tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
                        TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                        TEP_PRINT_NAME, TEP_PRINT_INFO);
        trace_seq_do_printf(&seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Simple tep example (simple-tep.c)

```
        sub_buf_size = tep_get_sub_buffer_size(tep);
        buf = malloc(sub_buf_size);
        read(fd, buf, sub_buf_size);
        kbuf = kbuffer_alloc(tep_get_header_page_size(tep) == 8, !tep_is_bigendian());
        kbuffer_load_subbuffer(kbuf, buf);
        record.data = kbuffer_read_event(kbuf, &ts);
        record.ts = ts;
        record.missed_events = kbuffer_missed_events(kbuf);
        record.size = kbuffer_event_size(kbuf);
        record.record_size = kbuffer_curr_size(kbuf);
        record.cpu = 0;
        kbuffer_next_event(kbuf, NULL);
        tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
                        TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                        TEP_PRINT_NAME, TEP_PRINT_INFO);
        trace_seq_do_printf(&seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Running simple-tep.c

```
# echo 1 > /sys/kernel/tracing/events/sched/sched_switch/enable
# cat /sys/kernel/tracing/per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C
```

Running simple-tep.c

```
# echo 1 > /sys/kernel/tracing/events/sched/sched_switch/enable
# cat /sys/kernel/tracing/per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
```

Running simple-tep.c

```
# echo 1 > /sys/kernel/tracing/events/sched/sched_switch/enable
# cat /sys/kernel/tracing/per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
8593.606023 <...>-2387 sched_switch prev_comm=cat prev_pid=2387 prev_prio=120 prev_state=Z ==> next_comm=swapper/0 next_pid=0 next_prio=120
```

Running simple-tep.c

```
# echo 1 > /sys/kernel/tracing/events/sched/sched_switch/enable
# cat /sys/kernel/tracing/per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
8593.606023 <...>-2387 sched_switch prev_comm=cat prev_pid=2387 prev_prio=120 prev_state=Z ==> next_comm=swapper/0 next_pid=0 next_prio=120
```

```
tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
                TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                TEP_PRINT_NAME, TEP_PRINT_INFO);
```

Running simple-tep.c

```
# echo 1 > /sys/kernel/tracing/events/sched/sched_switch/enable
# cat /sys/kernel/tracing/per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
8593.606023 <...>-2387 sched_switch prev_comm=cat prev_pid=2387 prev_prio=120 prev_state=Z ==> next_comm=swapper/0 next_pid=0 next_prio=120
```

```
tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
                TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                TEP_PRINT_NAME, TEP_PRINT_INFO);
```

Running simple-tep.c

%<decimal-shift>.<division>d

<decimal-shift> - How much to move the decimal to the left (6 places here) after division

<division> - What to divide the raw timestamp with before the decimal shift

```
# echo 1 > /sys/kernel/tracing/events/sched/sched_switch/enable
# cat /sys/kernel/tracing/per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
8593.606023 <...>-2387 sched_switch prev_comm=cat prev_pid=2387 prev_prio=120 prev_state=Z ==> next_comm=swapper/0 next_pid=0 next_prio=120
```

```
tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
                TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                TEP_PRINT_NAME, TEP_PRINT_INFO);
```

Running simple-tep.c

8593606022930 / 1000 = 8593606023   Move decimal 6 places: 8593.606023

```
# echo 1 > /sys/kernel/tracing/events/sched/sched_switch/enable
# cat /sys/kernel/tracing/per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
8593.606023 <...>-2387 sched_switch prev_comm=cat prev_pid=2387 prev_prio=120 prev_state=Z ==> next_comm=swapper/0 next_pid=0 next_prio=120
```

```
tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
                  TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                  TEP_PRINT_NAME, TEP_PRINT_INFO);
```

Running simple-tep.c

8593606022930 / 1 = 8593606022930   Move decimal 9 places: 8593.606029230

```
# echo 1 > /sys/kernel/tracing/events/sched/sched_switch/enable
# cat /sys/kernel/tracing/per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
8593.606022930 <...>-2387 sched_switch prev_comm=cat prev_pid=2387 prev_prio=120 prev_state=Z ==> next_comm=swapper/0 next_pid=0 next_prio=12
```

```
tep_print_event(tep, &seq, &record, "%9.1d %s-%d %s %s\n",
                TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                TEP_PRINT_NAME, TEP_PRINT_INFO);
```

Running simple-tep.c

Or just keep the raw timestamp

```
# echo 1 > /sys/kernel/tracing/events/sched/sched_switch/enable
# cat /sys/kernel/tracing/per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
8593606022930 <...>-2387 sched_switch prev_comm=cat prev_pid=2387 prev_prio=120 prev_state=Z ==> next_comm=swapper/0 next_pid=0 next_prio=120
```

```
tep_print_event(tep, &seq, &record, "%d %s-%d %s %s\n",
                TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                TEP_PRINT_NAME, TEP_PRINT_INFO);
```

Running simple-tep.c

```
# echo 1 > /sys/kernel/tracing/events/sched/sched_switch/enable
# cat /sys/kernel/tracing/per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
8593.606023 <...>-2387 sched_switch prev_comm=cat prev_pid=2387 prev_prio=120 prev_state=Z ==> next_comm=swapper/0 next_pid=0 next_prio=120
```

```
tep_print_event(tep, &seq, &record, "%6.100d %s-%d %s %s\n",
                  TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                  TEP_PRINT_NAME, TEP_PRINT_INFO);
```

Running simple-tep.c

```
# echo 1 > /sys/kernel/tracing/events/sched/sched_switch/enable
# cat /sys/kernel/tracing/per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
8593.606023 <...>-2387 sched_switch prev_comm=cat prev_pid=2387 prev_prio=120 prev_state=Z ==> next_comm=swapper/0 next_pid=0 next_prio=120
```

```
tep_print_event(tep, &seq, &record, "%6.100d %s-%d %s %s\n",
                  TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                  TEP_PRINT_NAME, TEP_PRINT_INFO);
```

Running simple-tep.c

```
# echo 1 > /sys/kernel/tracing/events/sched/sched_switch/enable
# cat /sys/kernel/tracing/per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
8593.606023 <...>-2387 sched_switch prev_comm=cat prev_pid=2387 prev_prio=120 prev_state=Z ==> next_comm=swapper/0 next_pid=0 next_prio=120
```

```
tep_print_event(tep, &seq, &record, "%6.100d %s-%d %s %s\n",
                  TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                  TEP_PRINT_NAME, TEP_PRINT_INFO);
```

Running simple-tep.c

```
# echo 1 > /sys/kernel/tracing/events/sched/sched_switch/enable
# cat /sys/kernel/tracing/per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
8593.606023 <...>-2387 sched_switch prev_comm=cat prev_pid=2387 prev_prio=120 prev_state=Z ==> next_comm=swapper/0 next_pid=0 next_prio=120
```

```
tep_print_event(tep, &seq, &record, "%6.100d %s-%d %s %s\n",
                  TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                  TEP_PRINT_NAME, TEP_PRINT_INFO);
```

Event formats

```
# cat events/sched/sched_switch/format
name: sched_switch
ID: 334
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), "|", { 0x0001, "S" }, { 0x0002, "D" }, { 0x0004, "T" }, { 0x0008, "t" }, {
0x0010, "X" }, { 0x0020, "Z" }, { 0x0040, "P" }, { 0x0080, "I" }) : "R", REC->prev_state & (((0x0000
| 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010 | 0x0020 | 0x0040) + 1) << 1) ? "+" : "",
REC->next_comm, REC->next_pid, REC->next_prio
```

Running simple-tep.c

```
# echo 1 > /sys/kernel/tracing/events/sched/sched_switch/enable
# cat /sys/kernel/tracing/per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
8593.606023 <...>-2387 sched_switch prev_comm=cat prev_pid=2387 prev_prio=120 prev_state=Z ==> next_comm=swapper/0 next_pid=0 next_prio=120
```

```
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), "|", { 0x0001, "S" }, { 0x0002, "D" },
{ 0x0004, "T" }, { 0x0008, "t" }, { 0x0010, "X" }, { 0x0020, "Z" }, { 0x0040, "P" }, { 0x0080, "I" }) : "R",
REC->prev_state & (((0x0000 | 0x0001 | 0x0002 | 0x0004 | 0x0008 | 0x0010 | 0x0020 | 0x0040) + 1) << 1) ? "+"
: "", REC->next_comm, REC->next_pid, REC->next_prio
```

saved_cmdlines

```
# head /sys/kernel/tracing/saved_cmdlines
5951 bash
5952 bash
5953 bash
5954 bash
5955 bash
16 rcu_preempt
5957 bash
5958 bash
5959 bash
5960 bash
```

saved_cmdlines

```
# head /sys/kernel/tracing/saved_cmdlines
5951 bash
5952 bash
5953 bash
5954 bash
5955 bash
16 rcu_preempt
5957 bash
5958 bash
5959 bash
5960 bash

# cat /sys/kernel/tracing/saved_cmdlines | wc -l
128
```

saved_cmdlines

```
# head /sys/kernel/tracing/saved_cmdlines
5951 bash
5952 bash
5953 bash
5954 bash
5955 bash
16 rcu_preempt
5957 bash
5958 bash
5959 bash
5960 bash

# cat /sys/kernel/tracing/saved_cmdlines | wc -l
128

# echo 512 > /sys/kernel/tracing/saved_cmdlines_size
```

Simple tep example (simple-tep.c)

```
int main(int argc, char **argv) {
      struct tep_record record;
      struct tep_handle *tep;
      struct kbuffer *kbuf;
      struct trace_seq seq;
      unsigned long long ts;
      void *buf;
      int sub_buf_size;
      int fd;

      fd = open(argv[1], O_RDONLY);

      trace_seq_init(&seq);
      tep = tep_alloc();
      buf = read_file("/sys/kernel/tracing/events/header_page");
      tep_parse_header_page(tep, buf, strlen(buf), 0);
      free(buf);
      buf = read_file("/sys/kernel/tracing/saved_cmdlines");
      tep_parse_saved_cmdlines(tep, buf);
      free(buf);
      buf = read_file("/sys/kernel/tracing/events/sched/sched_switch/format");
      tep_parse_event(tep, buf, strlen(buf), "sched");
      free(buf);
```

Using saved_cmdlines

```
# cd /sys/kernel/tracing
# echo > trace
# echo 1 > events/sched/sched_switch/enable
# sleep 5
# echo 0 > tracing_on
# cat per_cpu/cpu0/trace_pipe_raw > /tmp/raw0
^C
```

Using saved_cmdlines

```
# cd /sys/kernel/tracing
# echo > trace
# echo 1 > events/sched/sched_switch/enable
# sleep 5
# echo 0 > tracing_on
# dd if=per_cpu/cpu0/trace_pipe_raw iflag=nonblock of=/tmp/raw0
```

Using saved_cmdlines

```
# cd /sys/kernel/tracing
# echo > trace
# echo 1 > events/sched/sched_switch/enable
# sleep 5
# echo 0 > tracing_on
# dd if=per_cpu/cpu0/trace_pipe_raw iflag=nonblock of=/tmp/raw0

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
147467.800419 bash-5921 sched_switch prev_comm=bash prev_pid=5921 prev_prio=120 prev_state=S ==> next_comm=swa
```

Make helper function for reading record (simple-tep.c)

```c
	sub_buf_size = tep_get_sub_buffer_size(tep);
	buf = malloc(sub_buf_size);
	read(fd, buf, sub_buf_size);
	kbuf = kbuffer_alloc(tep_get_header_page_size(tep) == 8, !tep_is_bigendian());
	kbuffer_load_subbuffer(kbuf, buf);
	record.data = kbuffer_read_event(kbuf, &ts);
	record.ts = ts;
	record.missed_events = kbuffer_missed_events(kbuf);
	record.size = kbuffer_event_size(kbuf);
	record.record_size = kbuffer_curr_size(kbuf);
	record.cpu = 0;
	kbuffer_next_event(kbuf, NULL);
	tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
			TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
			TEP_PRINT_NAME, TEP_PRINT_INFO);
	trace_seq_do_printf(&seq);
	trace_seq_destroy(&seq);

	return 0;
}
```

Make helper function for reading record (simple-tep.c)

```
static void read_record(struct kbuffer *kbuf, struct tep_record *record)
{
	unsigned long long ts;

	record->data = kbuffer_read_event(kbuf, &ts);
	record->ts = ts;
	record->missed_events = kbuffer_missed_events(kbuf);
	record->size = kbuffer_event_size(kbuf);
	record->record_size = kbuffer_curr_size(kbuf);
	record->cpu = 0;
}
```

Make helper function for reading record (simple-tep.c)

```
      sub_buf_size = tep_get_sub_buffer_size(tep);
      buf = malloc(sub_buf_size);
      read(fd, buf, sub_buf_size);
      kbuf = kbuffer_alloc(tep_get_header_page_size(tep) == 8, !tep_is_bigendian());
      kbuffer_load_subbuffer(kbuf, buf);
      read_record(kbuf, &record);
      kbuffer_next_event(kbuf, NULL);
      tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
                  TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                  TEP_PRINT_NAME, TEP_PRINT_INFO);
      trace_seq_do_printf(&seq);
      trace_seq_destroy(&seq);

      return 0;
}
```

Read more than one record (simple-tep.c)

```
	sub_buf_size = tep_get_sub_buffer_size(tep);
	buf = malloc(sub_buf_size);
	read(fd, buf, sub_buf_size);
	kbuf = kbuffer_alloc(tep_get_header_page_size(tep) == 8, !tep_is_bigendian());
	kbuffer_load_subbuffer(kbuf, buf);
	read_record(kbuf, &record);
	kbuffer_next_event(kbuf, NULL);
	tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
			TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
			TEP_PRINT_NAME, TEP_PRINT_INFO);
	trace_seq_do_printf(&seq);
	trace_seq_destroy(&seq);

	return 0;
}
```

Read more than one record (simple-tep.c)

```
	sub_buf_size = tep_get_sub_buffer_size(tep);
	buf = malloc(sub_buf_size);
	read(fd, buf, sub_buf_size);
	kbuf = kbuffer_alloc(tep_get_header_page_size(tep) == 8, !tep_is_bigendian());
	kbuffer_load_subbuffer(kbuf, buf);
	for (;;) {
		read_record(kbuf, &record);
		kbuffer_next_event(kbuf, NULL);
		trace_seq_reset(&seq);
		tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
				TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
				TEP_PRINT_NAME, TEP_PRINT_INFO);
		trace_seq_do_printf(&seq);
	}
	trace_seq_destroy(&seq);
	return 0;
}
```

Multiple lines

```
# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
147467.800419 bash-5921 sched_switch prev_comm=bash prev_pid=5921 prev_prio=120 prev_state=S ==> next_comm=swapper/0 next_pid=0 next_prio=
147467.801061 <idle>-0 sched_switch prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=R ==> next_comm=rcu_preempt next_pid=16 next_p
147467.801073 rcu_preempt-16 sched_switch prev_comm=rcu_preempt prev_pid=16 prev_prio=120 prev_state=I ==> next_comm=swapper/0 next_pid=0
147467.805077 <idle>-0 sched_switch prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=R ==> next_comm=rcu_preempt next_pid=16 next_p
147467.805091 rcu_preempt-16 sched_switch prev_comm=rcu_preempt prev_pid=16 prev_prio=120 prev_state=I ==> next_comm=swapper/0 next_pid=0
147468.748077 <idle>-0 sched_switch prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=R ==> next_comm=rcu_preempt next_pid=16 next_p
147468.748090 rcu_preempt-16 sched_switch prev_comm=rcu_preempt prev_pid=16 prev_prio=120 prev_state=I ==> next_comm=swapper/0 next_pid=0
```

The function tracer

```
# echo function > current_tracer
# cat trace
# tracer: function
#
# entries-in-buffer/entries-written: 265996/1500091   #P:8
#
#                                _-----=> irqs-off/BH-disabled
#                               / _----=> need-resched
#                              | / _---=> hardirq/softirq
#                              || / _--=> preempt-depth
#                              ||| / _-=> migrate-disable
#                              |||| /     delay
#           TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#              | |         |   |||||     |         |
          <idle>-0       [006] d..2. 273461.685562: rcu_idle_exit <-cpuidle_enter_state
          <idle>-0       [006] d..3. 273461.685564: rcu_read_lock_sched_held <-trace_cpu_idle
          <idle>-0       [006] d..2. 273461.685564: sched_idle_set_state <-cpuidle_enter_state
          <idle>-0       [006] d..2. 273461.685565: __rcu_irq_enter_check_tick <-rcu_nmi_enter
          <idle>-0       [006] d..3. 273461.685566: rcu_read_lock_sched_held <-trace_hardirqs_off_finish
          <idle>-0       [006] d..2. 273461.685566: irq_enter_rcu <-sysvec_apic_timer_interrupt
          <idle>-0       [006] d..2. 273461.685566: preempt_count_add <-irq_enter_rcu
          <idle>-0       [006] d.h2. 273461.685567: tick_irq_enter <-irq_enter_rcu
          <idle>-0       [006] d.h2. 273461.685567: tick_check_oneshot_broadcast_this_cpu <-tick_irq_enter
          <idle>-0       [006] d.h2. 273461.685567: ktime_get <-tick_irq_enter
```

Function tracer format

```
# cat events/ftrace/function/format
name: function
ID: 1
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:unsigned long ip;	offset:8;	size:8;	signed:0;
	field:unsigned long parent_ip;	offset:16;	size:8;	signed:0;

print fmt: " %ps <-- %ps", (void *)REC->ip, (void *)REC->parent_ip
```

Function tracer format

```
# cat events/ftrace/function/format
name: function
ID: 1
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:unsigned long ip; offset:8;   size:8;     signed:0;
      field:unsigned long parent_ip;offset:16;  size:8;     signed:0;

print fmt: " %ps <-- %ps", (void *)REC->ip, (void *)REC->parent_ip
```

Add function information (simple-tep.c)

```
fd = open(argv[1], O_RDONLY);

trace_seq_init(&seq);
tep = tep_alloc();
buf = read_file("/sys/kernel/tracing/events/header_page");
tep_parse_header_page(tep, buf, strlen(buf), 0);
free(buf);
buf = read_file("/sys/kernel/tracing/saved_cmdlines");
tep_parse_saved_cmdlines(tep, buf);
free(buf);
buf = read_file("/sys/kernel/tracing/events/sched/sched_switch/format");
tep_parse_event(tep, buf, strlen(buf), "sched");
free(buf);
buf = read_file("/sys/kernel/tracing/events/ftrace/function/format");
tep_parse_event(tep, buf, strlen(buf), "ftrace");
free(buf);
```

Function tracing

```
# cd /sys/kernel/tracing
# echo > trace
# echo 0 > events/enable
# echo 1 > tracing_on
# echo function > current_tracer
# echo 0 > tracing_on
# dd if=per_cpu/cpu0/trace_pipe_raw iflag=nonblock of=/tmp/raw0
```

Function tracing

```
# cd /sys/kernel/tracing
# echo > trace
# echo 0 > events/enable
# echo 1 > tracing_on
# echo function > current_tracer
# echo 0 > tracing_on
# dd bs=4096 if=per_cpu/cpu0/trace_pipe_raw iflag=nonblock of=/tmp/raw0
```

Function tracing

```
# cd /sys/kernel/tracing
# echo > trace
# echo 0 > events/enable
# echo 1 > tracing_on
# echo function > current_tracer
# echo 0 > tracing_on
# dd bs=4096 if=per_cpu/cpu0/trace_pipe_raw iflag=nonblock of=/tmp/raw0

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
150199.029363 <idle>-0 function  0xffffffff98ca38e0 <-- 0xffffffff9819abc7
150199.029363 <idle>-0 function  0xffffffff98178140 <-- 0xffffffff98156350
150199.029363 <idle>-0 function  0xffffffff98159820 <-- 0xffffffff98ca3903
150199.029363 <idle>-0 function  0xffffffff98112a70 <-- 0xffffffff98ca391d
150199.029364 <idle>-0 function  0xffffffff98178140 <-- 0xffffffff9819ad1d
150199.029364 <idle>-0 function  0xffffffff981ae280 <-- 0xffffffff9819aae8
150199.029364 <idle>-0 function  0xffffffff9819d470 <-- 0xffffffff981ae2a6
150199.029364 <idle>-0 function  0xffffffff98178140 <-- 0xffffffff981566ee
150199.029365 <idle>-0 function  0xffffffff98178140 <-- 0xffffffff98156350
150199.029365 <idle>-0 function  0xffffffff981ae1e0 <-- 0xffffffff981ae2b4
150199.029365 <idle>-0 function  0xffffffff981ae040 <-- 0xffffffff981ae26f
150199.029365 <idle>-0 function  0xffffffff98ca3490 <-- 0xffffffff981ae065
[..]
```

Function tracing

```
# cd /sys/kernel/tracing
# echo > trace
# echo 0 > events/enable
# echo 1 > tracing_on
# echo function > current_tracer
# echo 0 > tracing_on
# dd bs=4096 if=per_cpu/cpu0/trace_pipe_raw iflag=nonblock of=/tmp/raw0

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
150199.029363 <idle>-0 function  0xffffffff98ca38e0 <-- 0xffffffff9819abc7
150199.029363 <idle>-0 function  0xffffffff98178140 <-- 0xffffffff98156350
150199.029363 <idle>-0 function  0xffffffff98159820 <-- 0xffffffff98ca3903
150199.029363 <idle>-0 function  0xffffffff98112a70 <-- 0xffffffff98ca391d
150199.029364 <idle>-0 function  0xffffffff98178140 <-- 0xffffffff9819ad1d
150199.029364 <idle>-0 function  0xffffffff981ae280 <-- 0xffffffff9819aae8
150199.029364 <idle>-0 function  0xffffffff9819d470 <-- 0xffffffff981ae2a6
150199.029364 <idle>-0 function  0xffffffff98178140 <-- 0xffffffff981566ee
150199.029365 <idle>-0 function  0xffffffff98178140 <-- 0xffffffff98156350
150199.029365 <idle>-0 function  0xffffffff981ae1e0 <-- 0xffffffff981ae2b4
150199.029365 <idle>-0 function  0xffffffff981ae040 <-- 0xffffffff981ae26f
150199.029365 <idle>-0 function  0xffffffff98ca3490 <-- 0xffffffff981ae065
[..]
```

Function tracing

```
# cd /sys/kernel/tracing
# echo > trace
# echo 0 > events/enable
# echo 1 > tracing_on
# echo function > current_tracer
# echo 0 > tracing_on
# dd bs=4096 if=per_cpu/cpu0/trace_pipe_raw iflag=nonblock of=/tmp/raw0

# gcc -o simple-tep -g -Wall simple-tep.c `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
150199.029363 <idle>-0 function  0xffffffff98ca38e0 <-- 0xffffffff9819abc7
150199.029363 <idle>-0 function  0xffffffff98178140 <-- 0xffffffff98156350
150199.029363 <idle>-0 function  0xffffffff9[illegible]9820 <-- 0xffffffff98[illegible]39[illegible]
150199.029363 <idle>-0 function  0xffffffff9[illegible]112a70 <-- 0xffffffff98ca39[illegible]
150199.029364 <idle>-0 function  0xffffffff9[illegible]178140 <-- 0xffffffff981[illegible]ad1d
150199.029364 <idle>-0 function  0xffffffff9[illegible]80 <-- 0xffffffff9[illegible]9aa[illegible]
150199.029364 <idle>-0 function  0xffffffff9819d470 <-- 0xffffffff981ae2a6
150199.029364 <idle>-0 function  0xffffffff98178140 <-- 0xffffffff981566ee
150199.029365 <idle>-0 function  0xffffffff98178140 <-- 0xffffffff98156350
150199.029365 <idle>-0 function  0xffffffff981ae1e0 <-- 0xffffffff981ae2b4
150199.029365 <idle>-0 function  0xffffffff981ae040 <-- 0xffffffff981ae26f
150199.029365 <idle>-0 function  0xffffffff98ca3490 <-- 0xffffffff981ae065
[..]
```

MOSTLY USELESS!

Function tracing

```
# cd /sys/kernel/tracing
# echo > trace
# echo 0 > events/enable
# echo 1 > tracing_on
# echo function > current_tracer
# echo 0 > tracing_on
# dd bs=4096 if=per_cpu/cpu0/trace_pipe_raw iflag=nonblock of=/tmp/raw0

# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
150199.029363 <idle>-0 function  0xffffffff98ca38e0 <-- 0xffffffff9819abc7
150199.029363 <idle>-0 function  0xffffffff98178140 <-- 0xffffffff98156350
150199.029363 <idle>-0 function  0xffffffff98159820 <-- 0xffffffff98ca3903
150199.029363 <idle>-0 function  0xffffffff98112a70 <-- 0xffffffff98ca391d
150199.029364 <i
150199.029364 <i
150199.029364 <idle>-0 function  0xffffffff9819d470 <-- 0xffffffff981ae2a6
150199.029364 <idle>-0 function  0xffffffff98178140 <-- 0xffffffff981566ee
150199.029365 <idle>-0 function  0xffffffff98178140 <-- 0xffffffff98156350
150199.029365 <idle>-0 function  0xffffffff981ae1e0 <-- 0xffffffff981ae2b4
150199.029365 <idle>-0 function  0xffffffff981ae040 <-- 0xffffffff981ae26f
150199.029365 <idle>-0 function  0xffffffff98ca3490 <-- 0xffffffff981ae065
[..]
```

```
print fmt: " %ps <-- %ps", (void *)REC->ip, (void *)REC->parent_ip
```

/proc/kallsyms

```
$ cat /proc/kallsyms | head
0000000000000000 T startup_64
0000000000000000 T _stext
0000000000000000 T _text
0000000000000000 T secondary_startup_64
0000000000000000 T secondary_startup_64_no_verify
0000000000000000 t verify_cpu
0000000000000000 T sev_verify_cbit
0000000000000000 T start_cpu0
0000000000000000 T __startup_64
0000000000000000 T startup_64_setup_env
```

/proc/kallsyms

```
$ cat /proc/kallsyms | head
0000000000000000 T startup_64
0000000000000000 T _stext
0000000000000000 T _text
0000000000000000 T secondary_startup_64
0000000000000000 T secondary_startup_64_no_verify
0000000000000000 t verify_cpu
0000000000000000 T sev_verify_cbit
0000000000000000 T start_cpu0
0000000000000000 T __startup_64
0000000000000000 T startup_64_setup_env
```

/proc/kallsyms

```
$ cat /proc/kallsyms | head
0000000000000000 T startup_64
0000000000000000 T _stext
0000000000000000 T _text
0000000000000000 T secondary_startup_64
0000000000000000 T secondary_startup_64_no_verify
0000000000000000 t verify_cpu
0000000000000000 T sev_verify_cbit
0000000000000000 T start_cpu0
0000000000000000 T __startup_64
0000000000000000 T startup_64_s[illegible]
```

TOTALLY USELESS!

/proc/kallsyms

```
$ cat /proc/kallsyms | head
0000000000000000 T startup_64
0000000000000000 T _stext
0000000000000000 T _text
0000000000000000 T secondary_startup_64
0000000000000000 T secondary_startup_64_no_verify
0000000000000000 t verify_cpu
0000000000000000 T sev_verify_cbit
0000000000000000 T start_cpu0
0000000000000000 T __startup_64
0000000000000000 T startup_64_setup_env

$ sudo cat /proc/kallsyms | head
ffffffff9f400000 T startup_64
ffffffff9f400000 T _stext
ffffffff9f400000 T _text
ffffffff9f400040 T secondary_startup_64
ffffffff9f400045 T secondary_startup_64_no_verify
ffffffff9f400110 t verify_cpu
ffffffff9f400210 T sev_verify_cbit
ffffffff9f400220 T start_cpu0
ffffffff9f400230 T __startup_64
ffffffff9f400470 T startup_64_setup_env
```

Add kallsyms information (simple-tep.c)

```
fd = open(argv[1], O_RDONLY);

trace_seq_init(&seq);
tep = tep_alloc();
buf = read_file("/sys/kernel/tracing/events/header_page");
tep_parse_header_page(tep, buf, strlen(buf), 0);
free(buf);
buf = read_file("/sys/kernel/tracing/saved_cmdlines");
tep_parse_saved_cmdlines(tep, buf);
free(buf);
buf = read_file("/sys/kernel/tracing/events/sched/sched_switch/format");
tep_parse_event(tep, buf, strlen(buf), "sched");
free(buf);
buf = read_file("/sys/kernel/tracing/events/ftrace/function/format");
tep_parse_event(tep, buf, strlen(buf), "ftrace");
free(buf);
buf = read_file("/proc/kallsyms");
tep_parse_kallsyms(tep, buf, strlen(buf), "ftrace");
free(buf);
```

Function tracing

```
# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
150199.029363 <idle>-0 function  _raw_spin_unlock_irqrestore <-- __hrtimer_run_queues
150199.029363 <idle>-0 function  rcu_read_lock_sched_held <-- lock_release
150199.029363 <idle>-0 function  do_raw_spin_unlock <-- _raw_spin_unlock_irqrestore
150199.029363 <idle>-0 function  preempt_count_sub <-- _raw_spin_unlock_irqrestore
150199.029364 <idle>-0 function  rcu_read_lock_sched_held <-- __hrtimer_run_queues
150199.029364 <idle>-0 function  tick_sched_timer <-- __hrtimer_run_queues
150199.029364 <idle>-0 function  ktime_get <-- tick_sched_timer
150199.029364 <idle>-0 function  rcu_read_lock_sched_held <-- lock_acquire
150199.029365 <idle>-0 function  rcu_read_lock_sched_held <-- lock_release
150199.029365 <idle>-0 function  tick_sched_do_timer <-- tick_sched_timer
150199.029365 <idle>-0 function  tick_do_update_jiffies64 <-- tick_sched_do_timer
150199.029365 <idle>-0 function  _raw_spin_lock <-- tick_do_update_jiffies64
150199.029365 <idle>-0 function  preempt_count_add <-- _raw_spin_lock
150199.029365 <idle>-0 function  rcu_read_lock_sched_held <-- lock_acquire
150199.029366 <idle>-0 function  do_raw_spin_trylock <-- _raw_spin_lock
150199.029366 <idle>-0 function  rcu_read_lock_sched_held <-- lock_acquired
150199.029366 <idle>-0 function  rcu_read_lock_sched_held <-- lock_acquire
150199.029366 <idle>-0 function  rcu_read_lock_sched_held <-- lock_release
150199.029366 <idle>-0 function  calc_global_load <-- tick_do_update_jiffies64
[..]
```

Function tracing (what about parent offsets?)

```
# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
150199.029363 <idle>-0 function  _raw_spin_unlock_irqrestore <-- __hrtimer_run_queues
150199.029363 <idle>-0 function  rcu_read_lock_sched_held <-- lock_release
150199.029363 <idle>-0 function  do_raw_spin_unlock <-- _raw_spin_unlock_irqrestore
150199.029363 <idle>-0 function  preempt_count_sub <-- _raw_spin_unlock_irqrestore
150199.029364 <idle>-0 function  rcu_read_lock_sched_held <-- __hrtimer_run_queues
150199.029364 <idle>-0 function  tick_sched_timer <-- __hrtimer_run_queues
150199.029364 <idle>-0 function  ktime_get <-- tick_sched_timer
150199.029364 <idle>-0 function  rcu_read_lock_sched_held <-- lock_acquire
150199.029365 <idle>-0 function  rcu_read_lock_sched_held <-- lock_release
150199.029365 <idle>-0 function  tick_sched_do_timer <-- tick_sched_timer
150199.029365 <idle>-0 function  tick_do_update_jiffies64 <-- tick_sched_do_timer
150199.029365 <idle>-0 function  _raw_spin_lock <-- tick_do_update_jiffies64
150199.029365 <idle>-0 function  preempt_count_add <-- _raw_spin_lock
150199.029365 <idle>-0 function  rcu_read_lock_sched_held <-- lock_acquire
150199.029366 <idle>-0 function  do_raw_spin_trylock <-- _raw_spin_lock
150199.029366 <idle>-0 function  rcu_read_lock_sched_held <-- lock_acquired
150199.029366 <idle>-0 function  rcu_read_lock_sched_held <-- lock_acquire
150199.029366 <idle>-0 function  rcu_read_lock_sched_held <-- lock_release
150199.029366 <idle>-0 function  calc_global_load <-- tick_do_update_jiffies64
[..]
```

Get parent offset (simple-tep.c)

```
int main(int argc, char **argv) {
      struct tep_record record;
      struct tep_handle *tep;
      struct tep_event *func_event;
      struct tep_format_field *func_ip;
      struct tep_format_field *func_pip;
      struct kbuffer *kbuf;
      struct trace_seq seq;
      unsigned long long ts;
      void *buf;
      int sub_buf_size;
      int fd;

      fd = open(argv[1], O_RDONLY);

      trace_seq_init(&seq);
      tep = tep_alloc();
      buf = read_file("/sys/kernel/tracing/events/header_page");
      tep_parse_header_page(tep, buf, strlen(buf), 0);
      free(buf);
      buf = read_file("/sys/kernel/tracing/saved_cmdlines");
      tep_parse_saved_cmdlines(tep, buf);
      free(buf);
```

Get parent offset (simple-tep.c)

```
buf = read_file("/sys/kernel/tracing/events/header_page");
tep_parse_header_page(tep, buf, strlen(buf), 0);
free(buf);
buf = read_file("/sys/kernel/tracing/saved_cmdlines");
tep_parse_saved_cmdlines(tep, buf);
free(buf);
buf = read_file("/sys/kernel/tracing/events/sched/sched_switch/format");
tep_parse_event(tep, buf, strlen(buf), "sched");
free(buf);
buf = read_file("/sys/kernel/tracing/events/ftrace/function/format");
tep_parse_event(tep, buf, strlen(buf), "ftrace");
free(buf);
func_event = tep_find_event_by_name(tep, "ftrace", "function");
func_ip = tep_find_field(func_event, "ip");
func_pip = tep_find_field(func_event, "parent_ip");
buf = read_file("/proc/kallsyms");
tep_parse_kallsyms(tep, buf, strlen(buf), "ftrace");
free(buf);
```

Get parent offset (simple-tep.c)

```
	for (;;) {
		read_record(kbuf, &record);
		if (!record.data)
			break;
		kbuffer_next_event(kbuf, NULL);
		trace_seq_reset(&seq);
		tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
				TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
				TEP_PRINT_NAME, TEP_PRINT_INFO);
		trace_seq_do_printf(&seq);
	}
	trace_seq_destroy(&seq);
	return 0;
}
```

Get parent offset (simple-tep.c)

```
	for (;;) {
		read_record(kbuf, &record);
		if (!record.data)
			break;
		kbuffer_next_event(kbuf, NULL);
		trace_seq_reset(&seq);
		if (tep_data_type(tep, &record) == func_event->id) {
			unsigned long long ip, pip;
			tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s ",
					TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
					TEP_PRINT_NAME);
			tep_read_number_field(func_ip, record.data, &ip);
			tep_read_number_field(func_pip, record.data, &pip);

			trace_seq_printf(&seq, "%s <-- %s+%lld\n",
					 tep_find_function(tep, ip),
					 tep_find_function(tep, pip),
					 pip - tep_find_function_address(tep, pip));
		} else {
			tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
					TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
					TEP_PRINT_NAME, TEP_PRINT_INFO);
		}
		trace_seq_do_printf(&seq);
	}
```

Function tracing with parent offset

```
# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
150199.029363 <idle>-0 function _raw_spin_unlock_irqrestore <-- __hrtimer_run_queues+407
150199.029363 <idle>-0 function rcu_read_lock_sched_held <-- lock_release+272
150199.029363 <idle>-0 function do_raw_spin_unlock <-- _raw_spin_unlock_irqrestore+35
150199.029363 <idle>-0 function preempt_count_sub <-- _raw_spin_unlock_irqrestore+61
150199.029364 <idle>-0 function rcu_read_lock_sched_held <-- __hrtimer_run_queues+749
150199.029364 <idle>-0 function tick_sched_timer <-- __hrtimer_run_queues+184
150199.029364 <idle>-0 function ktime_get <-- tick_sched_timer+38
150199.029364 <idle>-0 function rcu_read_lock_sched_held <-- lock_acquire+254
150199.029365 <idle>-0 function rcu_read_lock_sched_held <-- lock_release+272
150199.029365 <idle>-0 function tick_sched_do_timer <-- tick_sched_timer+52
150199.029365 <idle>-0 function tick_do_update_jiffies64 <-- tick_sched_do_timer+143
150199.029365 <idle>-0 function _raw_spin_lock <-- tick_do_update_jiffies64+37
150199.029365 <idle>-0 function preempt_count_add <-- _raw_spin_lock+21
150199.029365 <idle>-0 function rcu_read_lock_sched_held <-- lock_acquire+254
150199.029366 <idle>-0 function do_raw_spin_trylock <-- _raw_spin_lock+60
150199.029366 <idle>-0 function rcu_read_lock_sched_held <-- lock_acquired+286
150199.029366 <idle>-0 function rcu_read_lock_sched_held <-- lock_acquire+254
150199.029366 <idle>-0 function rcu_read_lock_sched_held <-- lock_release+272
150199.029366 <idle>-0 function calc_global_load <-- tick_do_update_jiffies64+194
[..]
```

Making it a handler (simple-tep.c)

```
	for (;;) {
		read_record(kbuf, &record);
		if (!record.data)
			break;
		kbuffer_next_event(kbuf, NULL);
		trace_seq_reset(&seq);
		if (tep_data_type(tep, &record) == func_event->id) {
			unsigned long long ip, pip;
			tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s ",
					TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
					TEP_PRINT_NAME);
			tep_read_number_field(func_ip, record.data, &ip);
			tep_read_number_field(func_pip, record.data, &pip);

			trace_seq_printf(&seq, "%s <-- %s+%lld\n",
					 tep_find_function(tep, ip),
					 tep_find_function(tep, pip),
					 pip - tep_find_function_address(tep, pip));
		} else {
			tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
					TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
					TEP_PRINT_NAME, TEP_PRINT_INFO);
		}
		trace_seq_do_printf(&seq);
	}
```

Making it a handler (simple-tep.c)

```
struct func_ips {
      struct tep_format_field       *ip;
      struct tep_format_field       *pip;
};

static int pfunc_index(struct trace_seq *seq, struct tep_record *record,
            struct tep_event *event, void *context)
{
      struct tep_handle *tep = event->tep;
      struct func_ips *fips = context;
      unsigned long long ip, pip;

      tep_read_number_field(fips->ip, record->data, &ip);
      tep_read_number_field(fips->pip, record->data, &pip);

      trace_seq_printf(seq, "%s <-- %s+%lld\n",
                  tep_find_function(tep, ip),
                  tep_find_function(tep, pip),
                  pip - tep_find_function_address(tep, pip));
      return 0;
}
```

Making it a handler (simple-tep.c)

```
int main(int argc, char **argv) {
      struct tep_record record;
      struct tep_handle *tep;
      struct tep_event *func_event;
      struct func_ips fips;
      struct kbuffer *kbuf;
      struct trace_seq seq;
      unsigned long long ts;
      void *buf;
      int sub_buf_size;
      int fd;

      fd = open(argv[1], O_RDONLY);

      trace_seq_init(&seq);
      tep = tep_alloc();
      buf = read_file("/sys/kernel/tracing/events/header_page");
      tep_parse_header_page(tep, buf, strlen(buf), 0);
      free(buf);
      buf = read_file("/sys/kernel/tracing/saved_cmdlines");
      tep_parse_saved_cmdlines(tep, buf);
      free(buf);
```

Making it a handler (simple-tep.c)

```
buf = read_file("/sys/kernel/tracing/events/header_page");
tep_parse_header_page(tep, buf, strlen(buf), 0);
free(buf);
buf = read_file("/sys/kernel/tracing/saved_cmdlines");
tep_parse_saved_cmdlines(tep, buf);
free(buf);
buf = read_file("/sys/kernel/tracing/events/sched/sched_switch/format");
tep_parse_event(tep, buf, strlen(buf), "sched");
free(buf);
buf = read_file("/sys/kernel/tracing/events/ftrace/function/format");
tep_parse_event(tep, buf, strlen(buf), "ftrace");
free(buf);
func_event = tep_find_event_by_name(tep, "ftrace", "function");
fips.ip = tep_find_field(func_event, "ip");
fips.pip = tep_find_field(func_event, "parent_ip");
tep_register_event_handler(tep, -1, "ftrace", "function", pfunc_index, &fips);
buf = read_file("/proc/kallsyms");
tep_parse_kallsyms(tep, buf, strlen(buf), "ftrace");
free(buf);
```

Making it a handler (simple-tep.c)

```
	sub_buf_size = tep_get_sub_buffer_size(tep);
	buf = malloc(sub_buf_size);
	read(fd, buf, sub_buf_size);
	kbuf = kbuffer_alloc(tep_get_header_page_size(tep) == 8, !tep_is_bigendian());
	kbuffer_load_subbuffer(kbuf, buf);
	for (;;) {
		read_record(kbuf, &record);
		kbuffer_next_event(kbuf, NULL);
		trace_seq_reset(&seq);
		tep_print_event(tep, &seq, &record, "%6.1000d %s-%d %s %s\n",
				TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
				TEP_PRINT_NAME, TEP_PRINT_INFO);
		trace_seq_do_printf(&seq);
	}
	trace_seq_destroy(&seq);
	return 0;
}
```

Function tracing with the handler

```
# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
150199.029363 <idle>-0 function _raw_spin_unlock_irqrestore <-- __hrtimer_run_queues+407
150199.029363 <idle>-0 function rcu_read_lock_sched_held <-- lock_release+272
150199.029363 <idle>-0 function do_raw_spin_unlock <-- _raw_spin_unlock_irqrestore+35
150199.029363 <idle>-0 function preempt_count_sub <-- _raw_spin_unlock_irqrestore+61
150199.029364 <idle>-0 function rcu_read_lock_sched_held <-- __hrtimer_run_queues+749
150199.029364 <idle>-0 function tick_sched_timer <-- __hrtimer_run_queues+184
150199.029364 <idle>-0 function ktime_get <-- tick_sched_timer+38
150199.029364 <idle>-0 function rcu_read_lock_sched_held <-- lock_acquire+254
150199.029365 <idle>-0 function rcu_read_lock_sched_held <-- lock_release+272
150199.029365 <idle>-0 function tick_sched_do_timer <-- tick_sched_timer+52
150199.029365 <idle>-0 function tick_do_update_jiffies64 <-- tick_sched_do_timer+143
150199.029365 <idle>-0 function _raw_spin_lock <-- tick_do_update_jiffies64+37
150199.029365 <idle>-0 function preempt_count_add <-- _raw_spin_lock+21
150199.029365 <idle>-0 function rcu_read_lock_sched_held <-- lock_acquire+254
150199.029366 <idle>-0 function do_raw_spin_trylock <-- _raw_spin_lock+60
150199.029366 <idle>-0 function rcu_read_lock_sched_held <-- lock_acquired+286
150199.029366 <idle>-0 function rcu_read_lock_sched_held <-- lock_acquire+254
150199.029366 <idle>-0 function rcu_read_lock_sched_held <-- lock_release+272
150199.029366 <idle>-0 function calc_global_load <-- tick_do_update_jiffies64+194
[..]
```

Simple tep plugin example (plugin-pfunc.c)

```
#include <event-parse.h>
#include <trace-seq.h>

struct func_ips {
      struct tep_format_field       *ip;
      struct tep_format_field       *pip;
};

static struct func_ips func_ips;

static int pfunc_index(struct trace_seq *seq, struct tep_record *record,
            struct tep_event *event, void *context)
{
      struct tep_handle *tep = event->tep;
      struct func_ips *fips = context;
      unsigned long long ip, pip;

      tep_read_number_field(fips->ip, record->data, &ip);
      tep_read_number_field(fips->pip, record->data, &pip);

      trace_seq_printf(seq, "%s <-- %s+%lld\n",
                   tep_find_function(tep, ip),
                   tep_find_function(tep, pip),
                   pip - tep_find_function_address(tep, pip));

      return 0;
}
```

Simple tep plugin example (plugin-pfunc.c)

```c
int TEP_PLUGIN_LOADER(struct tep_handle *tep)
{
      struct tep_event *func_event;

      func_event = tep_find_event_by_name(tep, "ftrace", "function");
      func_ips.ip = tep_find_field(func_event, "ip");
      func_ips.pip = tep_find_field(func_event, "parent_ip");
      tep_register_event_handler(tep, -1, "ftrace", "function", pfunc_index, &func_ips);
      return 0;
}

void TEP_PLUGIN_UNLOADER(struct tep_handle *tep)
{
      tep_unregister_event_handler(tep, -1, "ftrace", "function", pfunc_index, &func_ips);
}
```

Simple tep example (simple-tep.c)

```
buf = read_file("/sys/kernel/tracing/events/header_page");
tep_parse_header_page(tep, buf, strlen(buf), 0);
free(buf);
buf = read_file("/sys/kernel/tracing/saved_cmdlines");
tep_parse_saved_cmdlines(tep, buf);
free(buf);
buf = read_file("/sys/kernel/tracing/events/sched/sched_switch/format");
tep_parse_event(tep, buf, strlen(buf), "sched");
free(buf);
buf = read_file("/sys/kernel/tracing/events/ftrace/function/format");
tep_parse_event(tep, buf, strlen(buf), "ftrace");
free(buf);
func_event = tep_find_event_by_name(tep, "ftrace", "function");
fips.ip = tep_find_field(func_event, "ip");
fips.pip = tep_find_field(func_event, "parent_ip");
tep_register_event_handler(tep, -1, "ftrace", "function", pfunc_index, &fips);
buf = read_file("/proc/kallsyms");
tep_parse_kallsyms(tep, buf, strlen(buf), "ftrace");
free(buf);
```

Simple tep example (simple-tep.c)

```
buf = read_file("/sys/kernel/tracing/events/header_page");
tep_parse_header_page(tep, buf, strlen(buf), 0);
free(buf);
buf = read_file("/sys/kernel/tracing/saved_cmdlines");
tep_parse_saved_cmdlines(tep, buf);
free(buf);
buf = read_file("/sys/kernel/tracing/events/sched/sched_switch/format");
tep_parse_event(tep, buf, strlen(buf), "sched");
free(buf);
buf = read_file("/sys/kernel/tracing/events/ftrace/function/format");
tep_parse_event(tep, buf, strlen(buf), "ftrace");
free(buf);
tep_set_flag(tep, TEP_DISABLE_SYS_PLUGINS);
tep_load_plugins(tep);
buf = read_file("/proc/kallsyms");
tep_parse_kallsyms(tep, buf, strlen(buf), "ftrace");
free(buf);
```

Installing the plugin

```
# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# gcc -o plugin-pfunc.so -fPIC -shared -g -Wall plugin-pfunc.c  `pkg-config --cflags --libs libtraceevent`
```

Installing the plugin

```
# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# gcc -o plugin-pfunc.so -fPIC -shared -g -Wall plugin-pfunc.c  `pkg-config --cflags --libs libtraceevent`
# mkdir -p ~/.local/lib/traceevent/plugins
# mv plugin-pfunc.so ~/.local/lib/traceevent/plugins
```

Installing the plugin

```
# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# gcc -o plugin-pfunc.so -fPIC -shared -g -Wall plugin-pfunc.c  `pkg-config --cflags --libs libtraceevent`
# mkdir -p ~/.local/lib/traceevent/plugins
# mv plugin-pfunc.so ~/.local/lib/traceevent/plugins

# ./simple-tep /tmp/raw0
150199.029363 <idle>-0 function _raw_spin_unlock_irqrestore <-- __hrtimer_run_queues+407
150199.029363 <idle>-0 function rcu_read_lock_sched_held <-- lock_release+272
150199.029363 <idle>-0 function do_raw_spin_unlock <-- _raw_spin_unlock_irqrestore+35
150199.029363 <idle>-0 function preempt_count_sub <-- _raw_spin_unlock_irqrestore+61
150199.029364 <idle>-0 function rcu_read_lock_sched_held <-- __hrtimer_run_queues+749
150199.029364 <idle>-0 function tick_sched_timer <-- __hrtimer_run_queues+184
150199.029364 <idle>-0 function ktime_get <-- tick_sched_timer+38
150199.029364 <idle>-0 function rcu_read_lock_sched_held <-- lock_acquire+254
150199.029365 <idle>-0 function rcu_read_lock_sched_held <-- lock_release+272
150199.029365 <idle>-0 function tick_sched_do_timer <-- tick_sched_timer+52
150199.029365 <idle>-0 function tick_do_update_jiffies64 <-- tick_sched_do_timer+143
150199.029365 <idle>-0 function _raw_spin_lock <-- tick_do_update_jiffies64+37
150199.029365 <idle>-0 function preempt_count_add <-- _raw_spin_lock+21
150199.029365 <idle>-0 function rcu_read_lock_sched_held <-- lock_acquire+254
150199.029366 <idle>-0 function do_raw_spin_trylock <-- _raw_spin_lock+60
150199.029366 <idle>-0 function rcu_read_lock_sched_held <-- lock_acquired+286
[..]
```

Just to show the plugin worked

```
# rm ~/.local/lib/traceevent/plugins/plugin-pfunc.so
```

Just to show the plugin worked

```
# rm ~/.local/lib/traceevent/plugins/plugin-pfunc.so
# ./simple-tep /tmp/raw0
150199.029363 <idle>-0 function  _raw_spin_unlock_irqrestore <-- __hrtimer_run_queues
150199.029363 <idle>-0 function  rcu_read_lock_sched_held <-- lock_release
150199.029363 <idle>-0 function  do_raw_spin_unlock <-- _raw_spin_unlock_irqrestore
150199.029363 <idle>-0 function  preempt_count_sub <-- _raw_spin_unlock_irqrestore
150199.029364 <idle>-0 function  rcu_read_lock_sched_held <-- __hrtimer_run_queues
150199.029364 <idle>-0 function  tick_sched_timer <-- __hrtimer_run_queues
150199.029364 <idle>-0 function  ktime_get <-- tick_sched_timer
150199.029364 <idle>-0 function  rcu_read_lock_sched_held <-- lock_acquire
150199.029365 <idle>-0 function  rcu_read_lock_sched_held <-- lock_release
150199.029365 <idle>-0 function  tick_sched_do_timer <-- tick_sched_timer
150199.029365 <idle>-0 function  tick_do_update_jiffies64 <-- tick_sched_do_timer
150199.029365 <idle>-0 function  _raw_spin_lock <-- tick_do_update_jiffies64
150199.029365 <idle>-0 function  preempt_count_add <-- _raw_spin_lock
150199.029365 <idle>-0 function  rcu_read_lock_sched_held <-- lock_acquire
150199.029366 <idle>-0 function  do_raw_spin_trylock <-- _raw_spin_lock
150199.029366 <idle>-0 function  rcu_read_lock_sched_held <-- lock_acquired
[..]
```

Simple tep example (simple-tep.c)

```
buf = read_file("/sys/kernel/tracing/events/header_page");
tep_parse_header_page(tep, buf, strlen(buf), 0);
free(buf);
buf = read_file("/sys/kernel/tracing/saved_cmdlines");
tep_parse_saved_cmdlines(tep, buf);
free(buf);
buf = read_file("/sys/kernel/tracing/events/sched/sched_switch/format");
tep_parse_event(tep, buf, strlen(buf), "sched");
free(buf);
buf = read_file("/sys/kernel/tracing/events/ftrace/function/format");
tep_parse_event(tep, buf, strlen(buf), "ftrace");
free(buf);
tep_set_flag(tep, TEP_DISABLE_SYS_PLUGINS);
tep_load_plugins(tep);
buf = read_file("/proc/kallsyms");
tep_parse_kallsyms(tep, buf, strlen(buf), "ftrace");
free(buf);
```

Simple tep example (simple-tep.c)

```
buf = read_file("/sys/kernel/tracing/events/header_page");
tep_parse_header_page(tep, buf, strlen(buf), 0);
free(buf);
buf = read_file("/sys/kernel/tracing/saved_cmdlines");
tep_parse_saved_cmdlines(tep, buf);
free(buf);
buf = read_file("/sys/kernel/tracing/events/sched/sched_switch/format");
tep_parse_event(tep, buf, strlen(buf), "sched");
free(buf);
buf = read_file("/sys/kernel/tracing/events/ftrace/function/format");
tep_parse_event(tep, buf, strlen(buf), "ftrace");
free(buf);
tep_load_plugins(tep);
buf = read_file("/proc/kallsyms");
tep_parse_kallsyms(tep, buf, strlen(buf), "ftrace");
free(buf);
```

Just to show the plugin worked

```
# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
[..]
150199.029367 <idle>-0 function update_wall_time
150199.029367 <idle>-0 function    timekeeping_advance
150199.029368 <idle>-0 function       _raw_spin_lock_irqsave
150199.029368 <idle>-0 function preempt_count_add
150199.029368 <idle>-0 function rcu_read_lock_sched_held
150199.029368 <idle>-0 function do_raw_spin_trylock
150199.029368 <idle>-0 function rcu_read_lock_sched_held
150199.029368 <idle>-0 function       ntp_tick_length
150199.029369 <idle>-0 function       ntp_tick_length
150199.029369 <idle>-0 function rcu_read_lock_sched_held
150199.029369 <idle>-0 function       timekeeping_update
150199.029369 <idle>-0 function          ntp_get_next_leap
150199.029370 <idle>-0 function          update_vsyscall
150199.029370 <idle>-0 function          raw_notifier_call_chain
150199.029370 <idle>-0 function             __init_scratch_end
150199.029370 <idle>-0 function rcu_read_lock_sched_held
150199.029371 <idle>-0 function rcu_read_lock_sched_held
150199.029371 <idle>-0 function          update_fast_timekeeper
150199.029371 <idle>-0 function          update_fast_timekeeper
150199.029371 <idle>-0 function rcu_read_lock_sched_held
[..]
```

Simple tep example (simple-tep.c)

```
buf = read_file("/sys/kernel/tracing/events/header_page");
tep_parse_header_page(tep, buf, strlen(buf), 0);
free(buf);
buf = read_file("/sys/kernel/tracing/saved_cmdlines");
tep_parse_saved_cmdlines(tep, buf);
free(buf);
buf = read_file("/sys/kernel/tracing/events/sched/sched_switch/format");
tep_parse_event(tep, buf, strlen(buf), "sched");
free(buf);
buf = read_file("/sys/kernel/tracing/events/ftrace/function/format");
tep_parse_event(tep, buf, strlen(buf), "ftrace");
free(buf);
tep_load_plugins(tep);
tep_plugin_add_option("parent", "1");
buf = read_file("/proc/kallsyms");
tep_parse_kallsyms(tep, buf, strlen(buf), "ftrace");
free(buf);
```

Using options

```
# gcc -o simple-tep -g -Wall simple-tep.c  `pkg-config --cflags --libs libtraceevent`
# ./simple-tep /tmp/raw0
[..]
150199.029367 <idle>-0 function update_wall_time <-- tick_sched_do_timer
150199.029367 <idle>-0 function    timekeeping_advance <-- update_wall_time
150199.029368 <idle>-0 function       _raw_spin_lock_irqsave <-- timekeeping_advance
150199.029368 <idle>-0 function preempt_count_add <-- __raw_spin_lock_irqsave
150199.029368 <idle>-0 function rcu_read_lock_sched_held <-- lock_acquire
150199.029368 <idle>-0 function do_raw_spin_trylock <-- __raw_spin_lock_irqsave
150199.029368 <idle>-0 function rcu_read_lock_sched_held <-- lock_acquired
150199.029368 <idle>-0 function       ntp_tick_length <-- timekeeping_advance
150199.029369 <idle>-0 function       ntp_tick_length <-- timekeeping_advance
150199.029369 <idle>-0 function rcu_read_lock_sched_held <-- lock_acquire
150199.029369 <idle>-0 function       timekeeping_update <-- timekeeping_advance
150199.029369 <idle>-0 function          ntp_get_next_leap <-- timekeeping_update
150199.029370 <idle>-0 function          update_vsyscall <-- timekeeping_update
150199.029370 <idle>-0 function          raw_notifier_call_chain <-- timekeeping_update
150199.029370 <idle>-0 function             __init_scratch_end <-- raw_notifier_call_chain
150199.029370 <idle>-0 function rcu_read_lock_sched_held <-- lock_acquire
150199.029371 <idle>-0 function rcu_read_lock_sched_held <-- lock_release
150199.029371 <idle>-0 function          update_fast_timekeeper <-- timekeeping_update
150199.029371 <idle>-0 function          update_fast_timekeeper <-- timekeeping_update
150199.029371 <idle>-0 function rcu_read_lock_sched_held <-- lock_release
[..]
```

What options are there?

```
# trace-cmd list -O
  plugin:     ftrace
  option:     parent
    desc:     Print parent of functions for function events
     set:     0
------------
  plugin:     ftrace
  option:     indent
    desc:     Try to show function call indents, based on parents
     set:     1
------------
  plugin:     ftrace
  option:     offset
    desc:     Show function names as well as their offsets
     set:     0
============
  plugin:     fgraph
  option:     tailprint
    desc:     Print function name at function exit in function graph
     set:     0
------------
  plugin:     fgraph
  option:     depth
    desc:     Show the depth of each entry
     set:     0
```

libtracefs

- Deals with everything to do with the tracefs file system

libtracefs

- Deals with everything to do with the tracefs file system
- The most “mature” of the libraries

libtracefs

- Deals with everything to do with the tracefs file system
- The most “mature” of the libraries
- Depends on libtraceevent

libtracefs

- Deals with everything to do with the tracefs file system
- The most “mature” of the libraries
- Depends on libtraceevent
    - But has better interfaces

libtracefs

- Deals with everything to do with the tracefs file system
- The most “mature” of the libraries
- Depends on libtraceevent
  - But has better interfaces
  - A lot of work went into trying to fix the wrongs

libtracefs

- Deals with everything to do with the tracefs file system
- The most “mature” of the libraries
- Depends on libtraceevent
  - But has better interfaces
  - A lot of work went into trying to fix the wrongs
- This is the main library if you want to manage your own tracing

libtracefs

Some of the man page examples can be turned into executables!

```
$ cd Documentation
$ ls libtracefs-*.txt
libtracefs-dynevents.txt        libtracefs-instances-affinity.txt    libtracefs-sql.txt
libtracefs-eprobes.txt          libtracefs-instances-file-manip.txt  libtracefs-stream.txt
libtracefs-error.txt            libtracefs-instances-files.txt       libtracefs-synth2.txt
libtracefs-events-file.txt      libtracefs-instances-manage.txt      libtracefs-synth-info.txt
libtracefs-events-tep.txt       libtracefs-instances-utils.txt       libtracefs-synth.txt
libtracefs-events.txt           libtracefs-kprobes.txt               libtracefs-traceon.txt
libtracefs-files.txt            libtracefs-log.txt                   libtracefs-tracer.txt
libtracefs-filter.txt           libtracefs-marker_raw.txt            libtracefs.txt
libtracefs-function-filter.txt  libtracefs-marker.txt                libtracefs-uprobes.txt
libtracefs-hist-cont.txt        libtracefs-option-get.txt            libtracefs-utils.txt
libtracefs-hist-mod.txt         libtracefs-option-misc.txt
libtracefs-hist.txt             libtracefs-options.txt
```

libtracefs

Some of the man page examples can be turned into executables!

```
$ cd Documentation
$ ls libtracefs-*.txt
libtracefs-dynevents.txt        libtracefs-instances-affinity.txt    libtracefs-sql.txt
libtracefs-eprobes.txt          libtracefs-instances-file-manip.txt  libtracefs-stream.txt
libtracefs-error.txt            libtracefs-instances-files.txt       libtracefs-synth2.txt
libtracefs-events-file.txt      libtracefs-instances-manage.txt      libtracefs-synth-info.txt
libtracefs-events-tep.txt       libtracefs-instances-utils.txt       libtracefs-synth.txt
libtracefs-events.txt           libtracefs-kprobes.txt               libtracefs-traceon.txt
libtracefs-files.txt            libtracefs-log.txt                   libtracefs-tracer.txt
libtracefs-filter.txt           libtracefs-marker_raw.txt            libtracefs.txt
libtracefs-function-filter.txt  libtracefs-marker.txt                libtracefs-uprobes.txt
libtracefs-hist-cont.txt        libtracefs-option-get.txt            libtracefs-utils.txt
libtracefs-hist-mod.txt         libtracefs-option-misc.txt
libtracefs-hist.txt             libtracefs-options.txt

$ cd ..
$ make samples
```

libtracefs

Some of the man page examples can be turned into executables!

```
$ cd Documentation
$ ls libtracefs-*.txt
libtracefs-dynevents.txt        libtracefs-instances-affinity.txt    libtracefs-sql.txt
libtracefs-eprobes.txt          libtracefs-instances-file-manip.txt  libtracefs-stream.txt
libtracefs-error.txt            libtracefs-instances-files.txt       libtracefs-synth2.txt
libtracefs-events-file.txt      libtracefs-instances-manage.txt      libtracefs-synth-info.txt
libtracefs-events-tep.txt       libtracefs-instances-utils.txt       libtracefs-synth.txt
libtracefs-events.txt           libtracefs-kprobes.txt               libtracefs-traceon.txt
libtracefs-files.txt            libtracefs-log.txt                   libtracefs-tracer.txt
libtracefs-filter.txt           libtracefs-marker_raw.txt            libtracefs.txt
libtracefs-function-filter.txt  libtracefs-marker.txt                libtracefs-uprobes.txt
libtracefs-hist-cont.txt        libtracefs-option-get.txt            libtracefs-utils.txt
libtracefs-hist-mod.txt         libtracefs-option-misc.txt
libtracefs-hist.txt             libtracefs-options.txt

$ cd ..
$ make samples
$ ls bin
dynevents  filter           hist-cont           sqlhist  tracer
eprobes    function-filter  instances-affinity  stream   uprobes
error      hist             kprobes             synth
```

Simple tracefs example (simple-tracefs.c)

```c
#define _GNU_SOURCE
#include <stdlib.h>
#include <unistd.h>
#include <sched.h>
#include <tracefs.h>

static int callback(struct tep_event *event, struct tep_record *record,
                     int cpu, void *data)
{
        struct trace_seq *seq = data;

        trace_seq_reset(seq);
        tep_print_event(event->tep, seq, record, "%6.1000d %s-%d %s %s\n",
                        TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                        TEP_PRINT_NAME, TEP_PRINT_INFO);
        trace_seq_do_printf(seq);
        return 0;
}

int main(int argc, char **argv) {
        struct tep_handle *tep;
        struct trace_seq seq;
        cpu_set_t *cpus;

        cpus = CPU_ALLOC(1);
        CPU_ZERO_S(CPU_ALLOC_SIZE(1), cpus);
        CPU_SET_S(0, CPU_ALLOC_SIZE(1), cpus);

        tep = tracefs_local_events(NULL);

        trace_seq_init(&seq);
        tracefs_iterate_raw_events(tep, NULL, cpus, CPU_ALLOC_SIZE(1),
                                   callback, &seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Simple tracefs example (simple-tracefs.c)

```
#define _GNU_SOURCE
#include <stdlib.h>
#include <unistd.h>
#include <sched.h>
#include <tracefs.h>

static int callback(struct tep_event *event, struct tep_record *record,
                     int cpu, void *data)
{
        struct trace_seq *seq = data;

        trace_seq_reset(seq);
        tep_print_event(event->tep, seq, record, "%6.1000d %s-%d %s %s\n",
                        TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                        TEP_PRINT_NAME, TEP_PRINT_INFO);
        trace_seq_do_printf(seq);
        return 0;
}

int main(int argc, char **argv) {
        struct tep_handle *tep;
        struct trace_seq seq;
        cpu_set_t *cpus;

        cpus = CPU_ALLOC(1);
        CPU_ZERO_S(CPU_ALLOC_SIZE(1), cpus);
        CPU_SET_S(0, CPU_ALLOC_SIZE(1), cpus);

        tep = tracefs_local_events(NULL);

        trace_seq_init(&seq);
        tracefs_iterate_raw_events(tep, NULL, cpus, CPU_ALLOC_SIZE(1),
                                    callback, &seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Simple tracefs example (simple-tracefs.c)

```c
#define _GNU_SOURCE
#include <stdlib.h>
#include <unistd.h>
#include <sched.h>
#include <tracefs.h>

static int callback(struct tep_event *event, struct tep_record *record,
                    int cpu, void *data)
{
        struct trace_seq *seq = data;

        trace_seq_reset(seq);
        tep_print_event(event->tep, seq, record, "%6.1000d %s-%d %s %s\n",
                        TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                        TEP_PRINT_NAME, TEP_PRINT_INFO);
        trace_seq_do_printf(seq);
        return 0;
}

int main(int argc, char **argv) {
        struct tep_handle *tep;
        struct trace_seq seq;
        cpu_set_t *cpus;

        cpus = CPU_ALLOC(1);
        CPU_ZERO_S(CPU_ALLOC_SIZE(1), cpus);
        CPU_SET_S(0, CPU_ALLOC_SIZE(1), cpus);

        tep = tracefs_local_events(NULL);

        trace_seq_init(&seq);
        tracefs_iterate_raw_events(tep, NULL, cpus, CPU_ALLOC_SIZE(1),
                                   callback, &seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Simple tracefs example (simple-tracefs.c)

```
#define _GNU_SOURCE
#include <stdlib.h>
#include <unistd.h>
#include <sched.h>
#include <tracefs.h>

static int callback(struct tep_event *event, struct tep_record *record,
                    int cpu, void *data)
{
        struct trace_seq *seq = data;

        trace_seq_reset(seq);
        tep_print_event(event->tep, seq, record, "%6.1000d %s-%d %s %s\n",
                        TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                        TEP_PRINT_NAME, TEP_PRINT_INFO);
        trace_seq_do_printf(seq);
        return 0;
}

int main(int argc, char **argv) {
        struct tep_handle *tep;
        struct trace_seq seq;
        cpu_set_t *cpus;

        cpus = CPU_ALLOC(1);
        CPU_ZERO_S(CPU_ALLOC_SIZE(1), cpus);
        CPU_SET_S(0, CPU_ALLOC_SIZE(1), cpus);

        tep = tracefs_local_events(NULL);

        trace_seq_init(&seq);
        tracefs_iterate_raw_events(tep, NULL, cpus, CPU_ALLOC_SIZE(1),
                                   callback, &seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Simple tracefs example (simple-tracefs.c)

```
#define _GNU_SOURCE
#include <stdlib.h>
#include <unistd.h>
#include <sched.h>
#include <tracefs.h>

static int callback(struct tep_event *event, struct tep_record *record,
                     int cpu, void *data)
{
        struct trace_seq *seq = data;

        trace_seq_reset(seq);
        tep_print_event(event->tep, seq, record, "%6.1000d %s-%d %s %s\n",
                        TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                        TEP_PRINT_NAME, TEP_PRINT_INFO);
        trace_seq_do_printf(seq);
        return 0;
}

int main(int argc, char **argv) {
        struct tep_handle *tep;
        struct trace_seq seq;
        cpu_set_t *cpus;

        cpus = CPU_ALLOC(1);
        CPU_ZERO_S(CPU_ALLOC_SIZE(1), cpus);
        CPU_SET_S(0, CPU_ALLOC_SIZE(1), cpus);

        tep = tracefs_local_events(NULL);

        trace_seq_init(&seq);
        tracefs_iterate_raw_events(tep, NULL, cpus, CPU_ALLOC_SIZE(1),
                                   callback, &seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Using the libtracefs library

```
# cd /sys/kernel/tracing
# echo > trace
# echo 0 > events/enable
# echo 1 > tracing_on
# echo function > current_tracer
# echo 0 > tracing_on
```

Using the libtracefs library

```
# cd /sys/kernel/tracing
# echo > trace
# echo 0 > events/enable
# echo 1 > tracing_on
# echo function > current_tracer
# echo 0 > tracing_on

# gcc -o simple-tracefs -g -Wall simple-tracefs.c  `pkg-config --cflags --libs libtracefs`
```

Using the libtracefs library

```
# cd /sys/kernel/tracing
# echo > trace
# echo 0 > events/enable
# echo 1 > tracing_on
# echo function > current_tracer
# echo 0 > tracing_on

# gcc -o simple-tracefs -g -Wall simple-tracefs.c  `pkg-config --cflags --libs libtracefs`
# ./simple-tracefs
182768.918609 <idle>-0 function  do_raw_spin_trylock <-- __raw_spin_lock_irq
182768.918610 <idle>-0 function  rcu_read_lock_sched_held <-- lock_acquired
182768.918610 <idle>-0 function  enqueue_hrtimer <-- __hrtimer_run_queues
182768.918610 <idle>-0 function  rcu_read_lock_sched_held <-- enqueue_hrtimer
182768.918611 <idle>-0 function  hrtimer_update_next_event <-- hrtimer_interrupt
182768.918611 <idle>-0 function  __hrtimer_next_event_base <-- hrtimer_update_next_event
182768.918611 <idle>-0 function  __hrtimer_next_event_base <-- hrtimer_update_next_event
182768.918611 <idle>-0 function  _raw_spin_unlock_irqrestore <-- hrtimer_interrupt
182768.918611 <idle>-0 function  rcu_read_lock_sched_held <-- lock_release
182768.918612 <idle>-0 function  do_raw_spin_unlock <-- _raw_spin_unlock_irqrestore
182768.918612 <idle>-0 function  preempt_count_sub <-- _raw_spin_unlock_irqrestore
182768.918612 <idle>-0 function  tick_program_event <-- hrtimer_interrupt
182768.918612 <idle>-0 function  clockevents_program_event <-- hrtimer_interrupt
182768.918612 <idle>-0 function  ktime_get <-- clockevents_program_event
[..]
```

Using the libtracefs library

```
# cd /sys/kernel/tracing
# echo nop > current_tracer
# echo 1 > events/enable
# echo 1 > tracing_on
# echo 0 > tracing_on
```

Using the libtracefs library

```
# cd /sys/kernel/tracing
# echo nop > current_tracer
# echo 1 > events/enable
# echo 1 > tracing_on
# echo 0 > tracing_on

# ./simple-tracefs
184156.900367 <idle>-0 lock_acquired 0xffffffff9980bb58 jiffies_lock
184156.900367 <idle>-0 lock_acquire 0xffffffff9980bb08 jiffies_seq.seqcount
184156.900368 <idle>-0 lock_release 0xffffffff9980bb08 jiffies_seq.seqcount
184156.900368 <idle>-0 lock_release 0xffffffff9980bb58 jiffies_lock
184156.900368 <idle>-0 lock_acquire 0xffffffff9994f138 timekeeper_lock
184156.900369 <idle>-0 lock_acquired 0xffffffff9994f138 timekeeper_lock
184156.900369 <idle>-0 lock_acquire 0xffffffff9994efc8 tk_core.seq.seqcount
184156.900370 <idle>-0 lock_acquire 0xffffffffc0aff3e8 (null)
184156.900370 <idle>-0 lock_release 0xffffffffc0aff3e8 (null)
184156.900371 <idle>-0 lock_release 0xffffffff9994efc8 tk_core.seq.seqcount
184156.900371 <idle>-0 lock_release 0xffffffff9994f138 timekeeper_lock
184156.900372 <idle>-0 rcu_utilization [.Z<99><FF><FF><FF><FF>
184156.900373 <idle>-0 rcu_utilization H.Z<99><FF><FF><FF><FF>
184156.900374 <idle>-0 read_msr e8, value 79f1f2cb34e
184156.900374 <idle>-0 read_msr e7, value fe0f5dc0363
184156.900375 <idle>-0 lock_acquire 0xffff9e44d9df2e98 &rq->__lock
[..]
```

Using the libtracefs library

```
# cd /sys/kernel/tracing
# echo nop > current_tracer
# echo 1 > events/enable
# echo 1 > tracing_on
# echo 0 > tracing_on

# ./simple-tracefs
184156.900367 <idle>-0 lock_acquired 0xffffffff9980bb58 jiffies_lock
184156.900367 <idle>-0 lock_acquire 0xffffffff9980bb08 jiffies_seq.seqcount
184156.900368 <idle>-0 lock_release 0xffffffff9980bb08 jiffies_seq.seqcount
184156.900368 <idle>-0 lock_release 0xffffffff9980bb58 jiffies_lock
184156.900368 <idle>-0 lock_acquire 0xffffffff9994f138 timekeeper_lock
184156.900369 <idle>-0 lock_acquired 0xffffffff9994f138 timekeeper_lock
184156.900369 <idle>-0 lock_acquire 0xffffffff9994efc8 tk_core.seq.seqcount
184156.900370 <idle>-0 lock_acquire 0xffffffffc0aff3e8 (null)
184156.900370 <idle>-0 lock_release 0xffffffffc0aff3e8 (null)
184156.900371 <idle>-0 lock_release 0xffffffff9994efc8 tk_core.seq.seqcount
184156.900371 <idle>-0 lock_release 0xffffffff9994f138 timekeeper_lock
184156.900372 <idle>-0 rcu_utilization [.Z<99><FF><FF><FF><FF>
184156.900373 <idle>-0 rcu_utilization H.Z<99><FF><FF><FF><FF>
184156.900374 <idle>-0 read_msr e8, value 79f1f2cb34e
184156.900374 <idle>-0 read_msr e7, value fe0f5dc0363
184156.900375 <idle>-0 lock_acquire 0xffff9e44d9df2e98 &rq->__lock
[..]
```

Simple tracefs example (simple-tracefs.c)

```c
#define _GNU_SOURCE
#include <stdlib.h>
#include <unistd.h>
#include <sched.h>
#include <tracefs.h>

static int callback(struct tep_event *event, struct tep_record *record,
                    int cpu, void *data)
{
        struct trace_seq *seq = data;

        trace_seq_reset(seq);
        tep_print_event(event->tep, seq, record, "%6.1000d %s-%d %s %s\n",
                        TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                        TEP_PRINT_NAME, TEP_PRINT_INFO);
        trace_seq_do_printf(seq);
        return 0;
}

int main(int argc, char **argv) {
        struct tep_handle *tep;
        struct trace_seq seq;
        cpu_set_t *cpus;

        cpus = CPU_ALLOC(1);
        CPU_ZERO_S(CPU_ALLOC_SIZE(1), cpus);
        CPU_SET_S(0, CPU_ALLOC_SIZE(1), cpus);

        tep = tracefs_local_events(NULL);
        tep_set_long_size(tep, tep_get_header_page_size(tep));

        trace_seq_init(&seq);
        tracefs_iterate_raw_events(tep, NULL, cpus, CPU_ALLOC_SIZE(1),
                                   callback, &seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Simple tracefs example (simple-tracefs.c)

```
#define _GNU_SOURCE
#include <stdlib.h>
#include <unistd.h>
#include <sched.h>
#include <tracefs.h>

static int callback(struct tep_event *event, struct tep_record *record,
                    int cpu, void *data)
{
        struct trace_seq *seq = data;

        trace_seq_reset(seq);
        tep_print_event(event->tep, seq, record, "%6.1000d %s-%d %s %s\n",
                        TEP_PRINT_TIME, TEP_PRINT_COMM, TEP_PRINT_PID,
                        TEP_PRINT_NAME, TEP_PRINT_INFO);
```

Fixed by
https://patchwork.kernel.org/project/linux-trace-devel/patch/20220722142803.24919c8a@gandalf.local.home/

```
        struct tep_handle *tep;
        struct trace_seq seq;
        cpu_set_t *cpus;

        cpus = CPU_ALLOC(1);
        CPU_ZERO_S(CPU_ALLOC_SIZE(1), cpus);
        CPU_SET_S(0, CPU_ALLOC_SIZE(1), cpus);

        tep = tracefs_local_events(NULL);
        tep_set_long_size(tep, tep_get_header_page_size(tep));

        trace_seq_init(&seq);
        tracefs_iterate_raw_events(tep, NULL, cpus, CPU_ALLOC_SIZE(1),
                                   callback, &seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Using the libtracefs library

```
# cd /sys/kernel/tracing
# echo 0 > events/enable
# echo > trace
# echo 1 > events/rcu/enable
# echo 1 > tracing_on
# echo 0 > tracing_on

# gcc -o simple-tracefs -g -Wall simple-tracefs.c  `pkg-config --cflags --libs libtracefs`
# ./simple-tracefs
184573.497985 <idle>-0 rcu_utilization Start scheduler-tick
184573.497988 <idle>-0 rcu_utilization End scheduler-tick
184573.498987 <idle>-0 rcu_utilization Start scheduler-tick
184573.498989 <idle>-0 rcu_utilization End scheduler-tick
184573.498995 <idle>-0 rcu_utilization Start RCU core
184573.498996 <idle>-0 rcu_utilization End RCU core
184573.499995 <idle>-0 rcu_utilization Start scheduler-tick
184573.499998 <idle>-0 rcu_utilization End scheduler-tick
184573.500996 <idle>-0 rcu_utilization Start scheduler-tick
184573.500998 <idle>-0 rcu_utilization End scheduler-tick
184573.501996 <idle>-0 rcu_utilization Start scheduler-tick
184573.501999 <idle>-0 rcu_utilization End scheduler-tick
184573.502997 <idle>-0 rcu_utilization Start scheduler-tick
184573.502999 <idle>-0 rcu_utilization End scheduler-tick
[..]
```

Simple tracefs example (simple-tracefs.c)

```c
#define _GNU_SOURCE
#include <stdlib.h>
#include <unistd.h>
#include <sched.h>
#include <tracefs.h>

static int callback(struct tep_event *event, struct tep_record *record,
                     int cpu, void *data)
{
        struct trace_seq *seq = data;

        trace_seq_reset(seq);
        tep_print_event(event->tep, seq, record, "%6.1000d", TEP_PRINT_TIME);
        trace_seq_printf(seq, " [%03d] ", cpu);
        tep_print_event(event->tep, seq, record, "%s-%d %s %s\n",
                         TEP_PRINT_COMM, TEP_PRINT_PID,
                         TEP_PRINT_NAME, TEP_PRINT_INFO);
        trace_seq_do_printf(seq);
        return 0;
}

int main(int argc, char **argv) {
        struct tep_handle *tep;
        struct trace_seq seq;

        tep = tracefs_local_events(NULL);
        tep_set_long_size(tep, tep_get_header_page_size(tep));

        trace_seq_init(&seq);
        tracefs_iterate_raw_events(tep, NULL, NULL, 0, callback, &seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Using the libtracefs library

```
# cd /sys/kernel/tracing
# echo > trace
# echo 1 > events/enable
# echo 1 > tracing_on
# echo 0 > tracing_on

# gcc -o simple-tracefs -g -Wall simple-tracefs.c  `pkg-config --cflags --libs libtracefs`
# ./simple-tracefs
184877.908590 [006] <idle>-0 lock_release 0xffffffff9994efc8 tk_core.seq.seqcount
184877.908591 [006] <idle>-0 lock_acquire 0xffff9e44da9e3998 hrtimer_bases.lock
184877.908591 [006] <idle>-0 lock_acquired 0xffff9e44da9e3998 hrtimer_bases.lock
184877.908591 [006] <idle>-0 lock_release 0xffff9e44da9e3998 hrtimer_bases.lock
184877.908593 [006] <idle>-0 lock_acquire 0xffff9e44da9e3998 hrtimer_bases.lock
184877.908593 [006] <idle>-0 lock_acquired 0xffff9e44da9e3998 hrtimer_bases.lock
184877.908594 [006] <idle>-0 hrtimer_start hrtimer=0xffff9e44da9e4320 function=tick_sched_timer expires=184878
184877.908595 [006] <idle>-0 lock_acquire 0xffffffff9994efc8 read tk_core.seq.seqcount
184877.908595 [006] <idle>-0 lock_release 0xffffffff9994efc8 tk_core.seq.seqcount
184877.908595 [006] <idle>-0 write_msr 6e0, value 23a72c299f9cc
184877.908596 [006] <idle>-0 lock_release 0xffff9e44da9e3998 hrtimer_bases.lock
184877.908596 [006] <idle>-0 cpu_idle state=4 cpu_id=6
184877.918015 [002] <idle>-0 lock_acquire 0xffff9e44da1f2e98 &rq->__lock
184877.918016 [002] <idle>-0 lock_acquired 0xffff9e44da1f2e98 &rq->__lock
184877.918021 [002] <idle>-0 sched_wakeup comm=wpa_supplicant pid=1479 prio=120 target_cpu=002
[..]
```

Simple tracefs example (simple-tracefs.c)

```
int main(int argc, char **argv) {
        struct tep_handle *tep;
        struct trace_seq seq;

        tracefs_trace_off(NULL);
        tracefs_instance_file_write(NULL, "trace", "");
        tracefs_event_enable(NULL, NULL, NULL);
        tracefs_trace_on(NULL);
        sleep(1);
        tracefs_trace_off(NULL);

        tep = tracefs_local_events(NULL);
        tep_set_long_size(tep, tep_get_header_page_size(tep));

        trace_seq_init(&seq);
        tracefs_iterate_raw_events(tep, NULL, NULL, 0, callback, &seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Using the libtracefs library

```
# gcc -o simple-tracefs -g -Wall simple-tracefs.c  `pkg-config --cflags --libs libtracefs`
# ./simple-tracefs
186420.901189 [007] <idle>-0 lock_acquire 0xffffffff9994efc8 read tk_core.seq.seqcount
186420.901189 [007] <idle>-0 lock_release 0xffffffff9994efc8 tk_core.seq.seqcount
186420.901190 [007] <idle>-0 write_msr 6e0, value 23f3561f1ed14
186420.901190 [007] <idle>-0 lock_release 0xffff9e44dabe3998 hrtimer_bases.lock
186420.901191 [007] <idle>-0 cpu_idle state=4 cpu_id=7
186420.903052 [005] <idle>-0 cpu_idle state=4294967295 cpu_id=5
186420.903054 [005] <idle>-0 irq_enable caller=cpuidle_enter_state+0xef parent=0x0
186420.903056 [005] <idle>-0 irq_disable caller=irqentry_enter+0x47 parent=0x0
186420.903058 [005] <idle>-0 lock_acquire 0xffffffff9994efc8 read tk_core.seq.seqcount
186420.903058 [005] <idle>-0 lock_release 0xffffffff9994efc8 tk_core.seq.seqcount
186420.903059 [005] <idle>-0 lock_acquire 0xffffffff9980bb58 jiffies_lock
186420.903060 [005] <idle>-0 lock_acquired 0xffffffff9980bb58 jiffies_lock
186420.903060 [005] <idle>-0 lock_acquire 0xffffffff9980bb08 jiffies_seq.seqcount
186420.903061 [005] <idle>-0 lock_release 0xffffffff9980bb08 jiffies_seq.seqcount
186420.903061 [005] <idle>-0 lock_release 0xffffffff9980bb58 jiffies_lock
186420.903062 [005] <idle>-0 lock_acquire 0xffffffff9994f138 timekeeper_lock
186420.903062 [005] <idle>-0 lock_acquired 0xffffffff9994f138 timekeeper_lock
186420.903063 [005] <idle>-0 lock_acquire 0xffffffff9994efc8 tk_core.seq.seqcount
186420.903064 [005] <idle>-0 lock_acquire 0xffffffffc0aff3e8 (null)
186420.903064 [005] <idle>-0 lock_release 0xffffffffc0aff3e8 (null)
[..]
```

Filtering events

- `tracefs_event_filter_apply()`
  - Applies a filter string to an event

Filtering events

- `tracefs_event_filter_apply()`
  - Applies a filter string to an event
- `tracefs_event_filter_clear()`
  - Clears the filter of an event

Filtering events

- `tracefs_event_filter_apply()`
  - Applies a filter string to an event
- `tracefs_event_filter_clear()`
  - Clears the filter of an event
- `tracefs_filter_string_verify()`
  - Verifies a filter string works with a given event

Filtering events

- `tracefs_event_filter_apply()`
  - Applies a filter string to an event
- `tracefs_event_filter_clear()`
  - Clears the filter of an event
- `tracefs_filter_string_verify()`
  - Verifies a filter string works with a given event
- `tracefs_filter_string_append()`
  - Used to build up a string and verify along the way

Simple tracefs example (simple-tracefs.c)

```
int main(int argc, char **argv) {
        struct tep_handle *tep;
        struct trace_seq seq;
        struct tep_event *sched_switch;
        char filter[1024];
        int pid = getpid();
        char *err;

        tep = tracefs_local_events(NULL);
        tep_set_long_size(tep, tep_get_header_page_size(tep));

        sched_switch = tep_find_event_by_name(tep, "sched", "sched_switch");
        sprintf(filter, "prev_pid = %d || next_pid = %d", pid, pid);
        if (tracefs_filter_string_verify(sched_switch, filter, &err)) {
                printf("Failed filter\n%s\n", err);
                free(err);
                exit(-1);
        }
        tracefs_event_filter_apply(NULL, sched_switch, filter);

        tracefs_trace_off(NULL);
        tracefs_instance_file_write(NULL, "trace", "");
        tracefs_event_enable(NULL, NULL, NULL);
        tracefs_trace_on(NULL);
        sleep(1);
        tracefs_trace_off(NULL);
        tracefs_event_filter_clear(NULL, sched_switch);

        trace_seq_init(&seq);
        tracefs_iterate_raw_events(tep, NULL, NULL, 0, callback, &seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Filtering events

```
# gcc -o simple-tracefs -g -Wall simple-tracefs.c  `pkg-config --cflags --libs libtracefs`
# ./simple-tracefs
Failed filter
prev_pid = 11727 || next_pid = 11727
         ^
Invalid compare
```

Simple tracefs example (simple-tracefs.c)

```c
int main(int argc, char **argv) {
        struct tep_handle *tep;
        struct trace_seq seq;
        struct tep_event *sched_switch;
        char filter[1024];
        int pid = getpid();
        char *err;

        tep = tracefs_local_events(NULL);
        tep_set_long_size(tep, tep_get_header_page_size(tep));

        sched_switch = tep_find_event_by_name(tep, "sched", "sched_switch");
        sprintf(filter, "prev_pid = %d || next_pid = %d", pid, pid);
        if (tracefs_filter_string_verify(sched_switch, filter, &err)) {
                printf("Failed filter\n%s\n", err);
                free(err);
                exit(-1);
        }
        tracefs_event_filter_apply(NULL, sched_switch, filter);

        tracefs_trace_off(NULL);
        tracefs_instance_file_write(NULL, "trace", "");
        tracefs_event_enable(NULL, NULL, NULL);
        tracefs_trace_on(NULL);
        sleep(1);
        tracefs_trace_off(NULL);
        tracefs_event_filter_clear(NULL, sched_switch);

        trace_seq_init(&seq);
        tracefs_iterate_raw_events(tep, NULL, NULL, 0, callback, &seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Simple tracefs example (simple-tracefs.c)

```
int main(int argc, char **argv) {
        struct tep_handle *tep;
        struct trace_seq seq;
        struct tep_event *sched_switch;
        char filter[1024];
        int pid = getpid();
        char *err;

        tep = tracefs_local_events(NULL);
        tep_set_long_size(tep, tep_get_header_page_size(tep));

        sched_switch = tep_find_event_by_name(tep, "sched", "sched_switch");
        sprintf(filter, "prev_pid == %d || next_pid == %d", pid, pid);
        if (tracefs_filter_string_verify(sched_switch, filter, &err)) {
                printf("Failed filter\n%s\n", err);
                free(err);
                exit(-1);
        }
        tracefs_event_filter_apply(NULL, sched_switch, filter);

        tracefs_trace_off(NULL);
        tracefs_instance_file_write(NULL, "trace", "");
        tracefs_event_enable(NULL, NULL, NULL);
        tracefs_trace_on(NULL);
        sleep(1);
        tracefs_trace_off(NULL);
        tracefs_event_filter_clear(NULL, sched_switch);

        trace_seq_init(&seq);
        tracefs_iterate_raw_events(tep, NULL, NULL, 0, callback, &seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Filtering events

```
# gcc -o simple-tracefs -g -Wall simple-tracefs.c  `pkg-config --cflags --libs libtracefs`
# ./simple-tracefs
17365.834096 [004] <...>-12718 sched_stat_runtime comm=simple-tracefs pid=12718 runtime=437063 [ns] vruntime=45984470537 [ns]
17365.834100 [004] <...>-12718 sched_switch prev_comm=simple-tracefs prev_pid=12718 prev_prio=120 prev_state=S ==> next_comm=swapper/4 nex
17365.836664 [007] <idle>-0 sched_waking comm=rcu_preempt pid=16 prio=120 target_cpu=007
17365.836671 [007] <idle>-0 sched_wakeup comm=rcu_preempt pid=16 prio=120 target_cpu=007
17365.836691 [007] rcu_preempt-16 sched_stat_runtime comm=rcu_preempt pid=16 runtime=11153 [ns] vruntime=61922886106 [ns]
17365.837128 [007] <idle>-0 sched_waking comm=migration/7 pid=59 prio=0 target_cpu=007
17365.837130 [007] <idle>-0 sched_wakeup comm=migration/7 pid=59 prio=0 target_cpu=007
17365.840680 [007] <idle>-0 sched_waking comm=rcu_preempt pid=16 prio=120 target_cpu=007
17365.840687 [007] <idle>-0 sched_wakeup comm=rcu_preempt pid=16 prio=120 target_cpu=007
17365.840703 [007] rcu_preempt-16 sched_stat_runtime comm=rcu_preempt pid=16 runtime=8290 [ns] vruntime=61922894396 [ns]
17365.843682 [004] <idle>-0 sched_waking comm=kworker/4:2 pid=12601 prio=120 target_cpu=004
17365.843688 [004] <idle>-0 sched_wakeup comm=kworker/4:2 pid=12601 prio=120 target_cpu=004
[..]
17366.681670 [000] <idle>-0 sched_waking comm=in:imjournal pid=968 prio=120 target_cpu=000
17366.681677 [000] <idle>-0 sched_wakeup comm=in:imjournal pid=968 prio=120 target_cpu=000
17366.681722 [000] in:imjournal-968 sched_stat_runtime comm=in:imjournal pid=968 runtime=27729 [ns] vruntime=266248729 [ns]
17366.710685 [002] <idle>-0 sched_wake_idle_without_ipi cpu=3
17366.801667 [000] <idle>-0 sched_waking comm=vmware-usbarbit pid=1024 prio=120 target_cpu=000
17366.801675 [000] <idle>-0 sched_wakeup comm=vmware-usbarbit pid=1024 prio=120 target_cpu=000
17366.801704 [000] vmware-usbarbit-1024 sched_stat_runtime comm=vmware-usbarbit pid=1024 runtime=12576 [ns] vruntime=339539930 [ns]
17366.834151 [004] <idle>-0 sched_waking comm=simple-tracefs pid=12718 prio=120 target_cpu=004
17366.834160 [004] <idle>-0 sched_wakeup comm=simple-tracefs pid=12718 prio=120 target_cpu=004
17366.834171 [004] <idle>-0 sched_switch prev_comm=swapper/4 prev_pid=0 prev_prio=120 prev_state=R ==> next_comm=simple-tracefs
next_pid=12718 next_prio=120
```

Filtering events

```
# gcc -o simple-tracefs -g -Wall simple-tracefs.c  `pkg-config --cflags --libs libtracefs`
# ./simple-tracefs
17365.834096 [004] <...>-12718 sched_stat_runtime comm=simple-tracefs pid=12718 runtime=437063 [ns] vruntime=45984470537 [ns]
17365.834100 [004] <...>-12718 sched_switch prev_comm=simple-tracefs prev_pid=12718 prev_prio=120 prev_state=S ==> next_comm=swapper/4 nex
17365.836664 [007] <idle>-0 sched_waking comm=rcu_preempt pid=16 prio=120 target_cpu=007
17365.836671 [007] <idle>-0 sched_wakeup comm=rcu_preempt pid=16 prio=120 target_cpu=007
17365.836691 [007] rcu_preempt-16 sched_stat_runtime comm=rcu_preempt pid=16 runtime=11153 [ns] vruntime=61922886106 [ns]
17365.837128 [007] <idle>-0 sched_waking comm=migration/7 pid=59 prio=0 target_cpu=007
17365.837130 [007] <idle>-0 sched_wakeup comm=migration/7 pid=59 prio=0 target_cpu=007
17365.840680 [007] <idle>-0 sched_waking comm=rcu_preempt pid=16 prio=120 target_cpu=007
17365.840687 [007] <idle>-0 sched_wakeup comm=rcu_preempt pid=16 prio=120 target_cpu=007
17365.840703 [007] rcu_preempt-16 sched_stat_runtime comm=rcu_preempt pid=16 runtime=8290 [ns] vruntime=61922894396 [ns]
17365.843682 [004] <idle>-0 sched_waking comm=kworker/4:2 pid=12601 prio=120 target_cpu=004
17365.843688 [004] <idle>-0 sched_wakeup comm=kworker/4:2 pid=12601 prio=120 target_cpu=004
[..]
17366.681670 [000] <idle>-0 sched_waking comm=in:imjournal pid=968 prio=120 target_cpu=000
17366.681677 [000] <idle>-0 sched_wakeup comm=in:imjournal pid=968 prio=120 target_cpu=000
17366.681722 [000] in:imjournal-968 sched_stat_runtime comm=in:imjournal pid=968 runtime=27729 [ns] vruntime=266248729 [ns]
17366.710685 [002] <idle>-0 sched_wake_idle_without_ipi cpu=3
17366.801667 [000] <idle>-0 sched_waking comm=vmware-usbarbit pid=1024 prio=120 target_cpu=000
17366.801675 [000] <idle>-0 sched_wakeup comm=vmware-usbarbit pid=1024 prio=120 target_cpu=000
17366.801704 [000] vmware-usbarbit-1024 sched_stat_runtime comm=vmware-usbarbit pid=1024 runtime=12576 [ns] vruntime=339539930 [ns]
17366.834151 [004] <idle>-0 sched_waking comm=simple-tracefs pid=12718 prio=120 target_cpu=004
17366.834160 [004] <idle>-0 sched_wakeup comm=simple-tracefs pid=12718 prio=120 target_cpu=004
17366.834171 [004] <idle>-0 sched_switch prev_comm=swapper/4 prev_pid=0 prev_prio=120 prev_state=R ==> next_comm=simple-tracefs
next_pid=12718 next_prio=120
```

Simple tracefs example (simple-tracefs.c)

```
int main(int argc, char **argv) {
        struct tep_handle *tep;
        struct trace_seq seq;
        struct tep_event *sched_switch;
        char filter[1024];
        int pid = getpid();
        char *err;

        tep = tracefs_local_events(NULL);
        tep_set_long_size(tep, tep_get_header_page_size(tep));

        sched_switch = tep_find_event_by_name(tep, "sched", "sched_switch");
        sprintf(filter, "prev_pid == %d || next_pid == %d", pid, pid);
        if (tracefs_filter_string_verify(sched_switch, filter, &err)) {
                printf("Failed filter\n%s\n", err);
                free(err);
                exit(-1);
        }
        tracefs_event_filter_apply(NULL, sched_switch, filter);

        tracefs_trace_off(NULL);
        tracefs_instance_file_write(NULL, "trace", "");
        tracefs_event_enable(NULL, NULL, NULL);
        tracefs_trace_on(NULL);
        sleep(1);
        tracefs_trace_off(NULL);
        tracefs_load_cmdlines(NULL, tep);
        tracefs_event_filter_clear(NULL, sched_switch);

        trace_seq_init(&seq);
        tracefs_iterate_raw_events(tep, NULL, NULL, 0, callback, &seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Filtering events

```
# gcc -o simple-tracefs -g -Wall simple-tracefs.c  `pkg-config --cflags --libs libtracefs`
# ./simple-tracefs
 5435.656479 [004] simple-tracefs-11737 sched_stat_runtime comm=simple-tracefs pid=11737 runtime=328460 [ns] vruntime=44207303540 [ns]
 5435.656484 [004] simple-tracefs-11737 sched_switch prev_comm=simple-tracefs prev_pid=11737 prev_prio=120 prev_state=S ==> next_comm=swap
 5435.659155 [007] <idle>-0 sched_waking comm=rcu_preempt pid=16 prio=120 target_cpu=007
 5435.659161 [007] <idle>-0 sched_wakeup comm=rcu_preempt pid=16 prio=120 target_cpu=007
 5435.659182 [007] rcu_preempt-16 sched_stat_runtime comm=rcu_preempt pid=16 runtime=11511 [ns] vruntime=52930026869 [ns]
 5435.663172 [007] <idle>-0 sched_waking comm=rcu_preempt pid=16 prio=120 target_cpu=007
 5435.663180 [007] <idle>-0 sched_wakeup comm=rcu_preempt pid=16 prio=120 target_cpu=007
 5435.663200 [007] rcu_preempt-16 sched_stat_runtime comm=rcu_preempt pid=16 runtime=10465 [ns] vruntime=52930037334 [ns]
 5435.666171 [004] <idle>-0 sched_waking comm=kworker/4:0 pid=2333 prio=120 target_cpu=004
 5435.666181 [004] <idle>-0 sched_wakeup comm=kworker/4:0 pid=2333 prio=120 target_cpu=004
 5435.666203 [004] kworker/4:0-2333 sched_stat_runtime comm=kworker/4:0 pid=2333 runtime=8702 [ns] vruntime=49148958659 [ns]
 5435.668171 [004] <idle>-0 sched_waking comm=kworker/0:1 pid=14 prio=120 target_cpu=000
 5435.668179 [004] <idle>-0 sched_wake_idle_without_ipi cpu=0
 5435.668181 [004] <idle>-0 sched_wakeup comm=kworker/0:1 pid=14 prio=120 target_cpu=000
 5435.668197 [000] kworker/0:1-14 sched_stat_runtime comm=kworker/0:1 pid=14 runtime=19898 [ns] vruntime=38108623051 [ns]
 5435.740173 [006] <idle>-0 sched_waking comm=kcompactd0 pid=82 prio=120 target_cpu=006
 5435.740180 [006] <idle>-0 sched_wakeup comm=kcompactd0 pid=82 prio=120 target_cpu=006
[..]
 5436.600746 [004] <idle>-0 sched_wakeup comm=wpa_supplicant pid=1484 prio=120 target_cpu=004
 5436.600783 [004] wpa_supplicant-1484 sched_stat_runtime comm=wpa_supplicant pid=1484 runtime=27618 [ns] vruntime=9700326 [ns]
 5436.656534 [004] <idle>-0 sched_waking comm=simple-tracefs pid=11737 prio=120 target_cpu=004
 5436.656543 [004] <idle>-0 sched_wakeup comm=simple-tracefs pid=11737 prio=120 target_cpu=004
 5436.656553 [004] <idle>-0 sched_switch prev_comm=swapper/4 prev_pid=0 prev_prio=120 prev_state=R ==> next_comm=simple-tracefs
next_pid=11737 next_prio=120
```

Simple tracefs example (simple-tracefs.c)

```
int main(int argc, char **argv) {
        struct tep_handle *tep;
        struct trace_seq seq;
        struct tep_event *sched_switch;
        char *filter = NULL;
        char buf[24];
        int pid = getpid();

        tep = tracefs_local_events(NULL);
        tep_set_long_size(tep, tep_get_header_page_size(tep));

        sched_switch = tep_find_event_by_name(tep, "sched", "sched_switch");
        sprintf(buf, "%d", pid);
        errno = 0;
        tracefs_filter_string_append(sched_switch, &filter, TRACEFS_FILTER_COMPARE,
                                     "prev_pid", TRACEFS_COMPARE_EQ, buf);
        tracefs_filter_string_append(sched_switch, &filter, TRACEFS_FILTER_AND, NULL, 0, NULL);
        tracefs_filter_string_append(sched_switch, &filter, TRACEFS_FILTER_COMPARE,
                                     "next_pid", TRACEFS_COMPARE_EQ, buf);
        if (errno) {
                printf("Failed filter\n");
                exit(-1);
        }
        tracefs_event_filter_apply(NULL, sched_switch, filter);

        tracefs_trace_off(NULL);
        tracefs_instance_file_write(NULL, "trace", "");
        tracefs_event_enable(NULL, NULL, NULL);
        tracefs_trace_on(NULL);
        sleep(1);
        tracefs_trace_off(NULL);
        tracefs_event_filter_clear(NULL, sched_switch);

        trace_seq_init(&seq);
        tracefs_iterate_raw_events(tep, NULL, NULL, 0, callback, &seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Simple tracefs example (simple-tracefs.c)

```c
int main(int argc, char **argv) {
        struct tep_handle *tep;
        struct trace_seq seq;
        struct tep_event *sched_switch;
        char *filter = NULL;
        char buf[24];
        int pid = getpid();

        tep = tracefs_local_events(NULL);
        tep_set_long_size(tep, tep_get_header_page_size(tep));

        sched_switch = tep_find_event_by_name(tep, "sched", "sched_switch");
        sprintf(buf, "%d", pid);
        errno = 0;
        tracefs_filter_string_append(sched_switch, &filter, TRACEFS_FILTER_COMPARE,
                                      "prev_pid", TRACEFS_COMPARE_EQ, buf);
        tracefs_filter_string_append(sched_switch, &filter, TRACEFS_FILTER_AND, NULL, 0, NULL);
        tracefs_filter_string_append(sched_switch, &filter, TRACEFS_FILTER_COMPARE,
                                      "next_pid", TRACEFS_COMPARE_EQ, buf);
        if (errno) {
                printf("Failed filter\n");
                exit(-1);
        }
        tracefs_event_filter_apply(NULL, sched_switch, filter);

        tracefs_trace_off(NULL);
        tracefs_instance_file_write(NULL, "trace", "");
        tracefs_event_enable(NULL, NULL, NULL);
        tracefs_trace_on(NULL);
        sleep(1);
        tracefs_trace_off(NULL);
        tracefs_event_filter_clear(NULL, sched_switch);

        trace_seq_init(&seq);
        tracefs_iterate_raw_events(tep, NULL, NULL, 0, callback, &seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Simple tracefs example (simple-tracefs.c)

```c
int main(int argc, char **argv) {
        struct tep_handle *tep;
        struct trace_seq seq;
        struct tep_event *sched_switch;
        char *filter = NULL;
        char buf[24];
        int pid = getpid();

        tep = tracefs_local_events(NULL);
        tep_set_long_size(tep, tep_get_header_page_size(tep));

        sched_switch = tep_find_event_by_name(tep, "sched", "sched_switch");
        sprintf(buf, "%d", pid);
        errno = 0;
        tracefs_filter_string_append(sched_switch, &filter, TRACEFS_FILTER_COMPARE
```

Fixed by
https://patchwork.kernel.org/project/linux-trace-devel/patch/20220722161732.0c5d7023@gandalf.local.home/

```c
                printf("Failed filter\n");
                exit(-1);
        }
        tracefs_event_filter_apply(NULL, sched_switch, filter);

        tracefs_trace_off(NULL);
        tracefs_instance_file_write(NULL, "trace", "");
        tracefs_event_enable(NULL, NULL, NULL);
        tracefs_trace_on(NULL);
        sleep(1);
        tracefs_trace_off(NULL);
        tracefs_event_filter_clear(NULL, sched_switch);

        trace_seq_init(&seq);
        tracefs_iterate_raw_events(tep, NULL, NULL, 0, callback, &seq);
        trace_seq_destroy(&seq);

        return 0;
}
```

Function filtering

- **tracefs_function_filter**()
  - Only trace functions in this list

Function filtering

- **tracefs_function_filter**()
  - Only trace functions in this list
- **tracefs_function_notrace**()
  - Do not trace functions in this list
  - Do not trace even if it is in tracefs_function_filter()

Function filtering

- **tracefs_function_filter**()
  - Only trace functions in this list
- **tracefs_function_notrace**()
  - Do not trace functions in this list
  - Do not trace even if it is in tracefs_function_filter()
- **tracefs_filter_functions**()
  - List the possible functions to filter

Simple tracefs example (simple-tracefs.c)

```
int main(int argc, char **argv) {
      struct tep_handle *tep;
      struct trace_seq seq;

      tep = tracefs_local_events(NULL);
      tep_set_long_size(tep, tep_get_header_page_size(tep));

      tracefs_trace_off(NULL);
      tracefs_function_filter(NULL, "*lock*", NULL, TRACEFS_FL_RESET | TRACEFS_FL_CONTINUE);
      tracefs_function_notrace(NULL, "*clock*", NULL, 0);
      tracefs_tracer_set(NULL, TRACEFS_TRACER_FUNCTION);
      tracefs_trace_on(NULL);
      sleep(1);
      tracefs_trace_off(NULL);

      trace_seq_init(&seq);
      tracefs_iterate_raw_events(tep, NULL, NULL, 0, callback, &seq);
      trace_seq_destroy(&seq);
      tracefs_tracer_clear(NULL);

      return 0;
}
```

Simple tracefs example (simple-tracefs.c)

```
int main(int argc, char **argv) {
      struct tep_handle *tep;
      struct trace_seq seq;

      tep = tracefs_local_events(NULL);
      tep_set_long_size(tep, tep_get_header_page_size(tep));

      tracefs_trace_off(NULL);
      tracefs_function_filter(NULL, "*lock*", NULL, TRACEFS_FL_RESET | TRACEFS_FL_CONTINUE);
      tracefs_function_notrace(NULL, "*clock*", NULL, 0);
      tracefs_tracer_set(NULL, TRACEFS_TRACER_CUSTOM, “function”);
      tracefs_trace_on(NULL);
      sleep(1);
      tracefs_trace_off(NULL);

      trace_seq_init(&seq);
      tracefs_iterate_raw_events(tep, NULL, NULL, 0, callback, &seq);
      trace_seq_destroy(&seq);
      tracefs_tracer_clear(NULL);

      return 0;
}
```

Filtering functions

```
# gcc -o simple-tracefs -g -Wall simple-tracefs.c  `pkg-config --cflags --libs libtracefs`
# ./simple-tracefs
22263.276473 [001] simple-tracefs-13186 function  mutex_unlock <-- rb_simple_write
22263.276474 [001] simple-tracefs-13186 function  __mutex_unlock_slowpath <-- rb_simple_write
22263.276474 [001] simple-tracefs-13186 function  rcu_read_lock_sched_held <-- lock_release
22263.276474 [001] simple-tracefs-13186 function  rcu_read_lock_sched_held <-- lock_release
22263.276474 [001] simple-tracefs-13186 function  rcu_read_lock_any_held <-- vfs_write
22263.276475 [001] simple-tracefs-13186 function  rcu_read_lock_sched_held <-- trace_hardirqs_on_prepare
22263.276476 [001] simple-tracefs-13186 function  rcu_read_lock_sched_held <-- trace_hardirqs_off_finish
22263.276476 [001] simple-tracefs-13186 function  _raw_spin_lock <-- close_fd
22263.276476 [001] simple-tracefs-13186 function  rcu_read_lock_sched_held <-- lock_acquire
22263.276476 [001] simple-tracefs-13186 function  do_raw_spin_trylock <-- _raw_spin_lock
22263.276476 [001] simple-tracefs-13186 function  rcu_read_lock_sched_held <-- lock_acquired
22263.276476 [001] simple-tracefs-13186 function  _raw_spin_unlock <-- close_fd
22263.276476 [001] simple-tracefs-13186 function  rcu_read_lock_sched_held <-- lock_release
22263.276477 [001] simple-tracefs-13186 function  do_raw_spin_unlock <-- _raw_spin_unlock
22263.276477 [001] simple-tracefs-13186 function  locks_remove_posix <-- filp_close
22263.276477 [001] simple-tracefs-13186 function  _raw_spin_lock_irq <-- task_work_run
22263.276477 [001] simple-tracefs-13186 function  rcu_read_lock_sched_held <-- lock_acquire
22263.276477 [001] simple-tracefs-13186 function  do_raw_spin_trylock <-- __raw_spin_lock_irq
22263.276477 [001] simple-tracefs-13186 function  rcu_read_lock_sched_held <-- lock_acquired
22263.276477 [001] simple-tracefs-13186 function  _raw_spin_unlock_irq <-- task_work_run
22263.276477 [001] simple-tracefs-13186 function  rcu_read_lock_sched_held <-- lock_release
22263.276477 [001] simple-tracefs-13186 function  do_raw_spin_unlock <-- _raw_spin_unlock_irq
22263.276478 [001] simple-tracefs-13186 function  locks_remove_file <-- __fput
[..]
```

Simple tracefs example (simple-tracefs.c)

```
int main(int argc, char **argv) {
      struct tep_handle *tep;
      struct trace_seq seq;

      tep = tracefs_local_events(NULL);
      tep_set_long_size(tep, tep_get_header_page_size(tep));

      tracefs_trace_off(NULL);
      tracefs_function_filter(NULL, ".*\\(do_\\)\\?raw.*lock.*", NULL, TRACEFS_FL_RESET);
      tracefs_tracer_set(NULL, TRACEFS_TRACER_FUNCTION);
      tracefs_trace_on(NULL);
      sleep(1);
      tracefs_trace_off(NULL);

      trace_seq_init(&seq);
      tracefs_iterate_raw_events(tep, NULL, NULL, 0, callback, &seq);
      trace_seq_destroy(&seq);
      tracefs_tracer_clear(NULL);

      return 0;
}
```

Filtering functions

```
# gcc -o simple-tracefs -g -Wall simple-tracefs.c  `pkg-config --cflags --libs libtracefs`
# ./simple-tracefs
23668.713403 [000] simple-tracefs-13365 function  _raw_spin_lock <-- close_fd
23668.713404 [000] simple-tracefs-13365 function  do_raw_spin_trylock <-- _raw_spin_lock
23668.713404 [000] simple-tracefs-13365 function  _raw_spin_unlock <-- close_fd
23668.713404 [000] simple-tracefs-13365 function  do_raw_spin_unlock <-- _raw_spin_unlock
23668.713404 [000] simple-tracefs-13365 function  _raw_spin_lock_irq <-- task_work_run
23668.713405 [000] simple-tracefs-13365 function  do_raw_spin_trylock <-- __raw_spin_lock_irq
23668.713405 [000] simple-tracefs-13365 function  _raw_spin_unlock_irq <-- task_work_run
23668.713405 [000] simple-tracefs-13365 function  do_raw_spin_unlock <-- _raw_spin_unlock_irq
23668.713406 [000] simple-tracefs-13365 function  _raw_spin_lock <-- lockref_put_or_lock
23668.713406 [000] simple-tracefs-13365 function  do_raw_spin_trylock <-- _raw_spin_lock
23668.713406 [000] simple-tracefs-13365 function  _raw_spin_unlock <-- lockref_put_or_lock
23668.713406 [000] simple-tracefs-13365 function  do_raw_spin_unlock <-- _raw_spin_unlock
23668.713410 [000] simple-tracefs-13365 function  _raw_spin_lock_irqsave <-- hrtimer_start_range_ns
23668.713410 [000] simple-tracefs-13365 function  do_raw_spin_trylock <-- __raw_spin_lock_irqsave
23668.713410 [000] simple-tracefs-13365 function  _raw_spin_unlock_irqrestore <-- do_nanosleep
23668.713411 [000] simple-tracefs-13365 function  do_raw_spin_unlock <-- _raw_spin_unlock_irqrestore
23668.713411 [000] simple-tracefs-13365 function  _raw_spin_lock_nested <-- __schedule
23668.713411 [000] simple-tracefs-13365 function  do_raw_spin_trylock <-- _raw_spin_lock_nested
23668.713414 [000] <idle>-0 function  _raw_spin_unlock <-- finish_task_switch.isra.0
23668.713414 [000] <idle>-0 function  do_raw_spin_unlock <-- _raw_spin_unlock
23668.714260 [000] <idle>-0 function  _raw_spin_lock_irqsave <-- hrtimer_interrupt
23668.714261 [000] <idle>-0 function  do_raw_spin_trylock <-- __raw_spin_lock_irqsave
23668.714263 [000] <idle>-0 function  _raw_spin_unlock_irqrestore <-- __hrtimer_run_queues
23668.714263 [000] <idle>-0 function  do_raw_spin_unlock <-- _raw_spin_unlock_irqrestore
[..]
```

Simple tracefs example (filter-functions.c)

```c
#include <stdio.h>
#include <stdlib.h>
#include <tracefs.h>

int main(int argc, char **argv) {
	char *modules = NULL;
	char **list = NULL;
	int ret, i;

	if (argc < 2) {
		printf("usage: filter-functions regex [module]\n");
		exit(-1);
	}
	if (argc > 2)
		modules = argv[2];

	ret = tracefs_filter_functions(argv[1], modules, &list);
	if (ret < 0)
		exit(-1);
	for (i = 0; list && list[i]; i++)
		printf("%s\n", list[i]);
	tracefs_list_free(list);

	return 0;
}
```

Filtering functions

```
# gcc -o filter-functions -g -Wall filter-functions.c  `pkg-config --cflags --libs libtracefs`
# ./filter-functions '.*\(do_\)\?raw.*lock'
raw_spin_rq_trylock
raw_spin_rq_unlock
do_raw_spin_lock
do_raw_spin_trylock
do_raw_spin_unlock
do_raw_read_lock
do_raw_read_trylock
do_raw_read_unlock
do_raw_write_lock
do_raw_write_trylock
do_raw_write_unlock
regmap_lock_raw_spinlock
regmap_unlock_raw_spinlock
_raw_spin_lock
_raw_spin_lock_nest_lock
_raw_spin_trylock
_raw_spin_unlock
_raw_read_trylock
_raw_read_lock
_raw_write_trylock
_raw_write_lock
_raw_read_unlock
_raw_write_unlock
```

Filtering functions

```
# ./filter-functions '*debug*' drm
drm_mode_debug_printmodeline
drm_atomic_debugfs_init
drm_framebuffer_debugfs_init
__drm_printfn_debug
drm_client_debugfs_internal_clients
drm_client_debugfs_init
drm_debugfs_open
drm_debugfs_create_files
drm_debugfs_remove_files
drm_debugfs_init
drm_debugfs_cleanup
drm_debugfs_connector_add
drm_debugfs_connector_remove
drm_debugfs_crtc_add
drm_debugfs_crtc_remove
drm_debugfs_crtc_crc_add
```

Something more useful

```
# ./filter-functions '*common_interrupt*'
__common_interrupt
```

Something more useful

```
# ./filter-functions '*common_interrupt*'
__common_interrupt

# trace-cmd list -e ‘ftrace:f.*’
ftrace:function
ftrace:funcgraph_entry
ftrace:funcgraph_exit
ftrace:func_repeats
```

Something more useful

```
# ./filter-functions '*common_interrupt*'
__common_interrupt

# trace-cmd list -e ‘ftrace:f.*’
ftrace:function
ftrace:funcgraph_entry
ftrace:funcgraph_exit
ftrace:func_repeats

# trace-cmd list -e irq_handler_entry -F
system: irq
name: irq_handler_entry
ID: 142
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:int irq;	offset:8;	size:4;	signed:1;
	field:__data_loc char[] name;	offset:12;	size:4;	signed:1;
```

Simple tracefs example (irq-lat.c)

```c
#include <stdlib.h>
#include <unistd.h>
#include <tracefs.h>

struct cpu_info {
	unsigned long long	start;
	unsigned long long	irq_vec;
	char			*irq_name;
};

struct irq_context {
	struct tep_handle	*tep;
	struct tep_event	*func_enter;
	struct tep_event	*func_exit;
	struct tep_event	*irq;
	struct tep_format_field	*irq_vec;
	struct tep_format_field	*irq_name;
	struct cpu_info		max;
	int			nr_cpus;
	struct cpu_info		*cpus;
	struct trace_seq	seq;
	int			max_cpu;
};

static int callback(struct tep_event *event, struct tep_record *record, int cpu, void *data);

int main (int argc, char **argv)
{
	struct irq_context context;

	context.tep = tracefs_local_events(NULL);
	tep_set_cpus(context.tep, sysconf(_SC_NPROCESSORS_CONF));
	context.nr_cpus = tep_get_cpus(context.tep);
	context.cpus = calloc(context.nr_cpus, sizeof(struct cpu_info));
	context.func_enter = tep_find_event_by_name(context.tep, "ftrace", "funcgraph_entry");
	context.func_exit = tep_find_event_by_name(context.tep, "ftrace", "funcgraph_exit");
	context.irq = tep_find_event_by_name(context.tep, "irq", "irq_handler_entry");
	context.irq_vec = tep_find_field(context.irq, "irq");
	context.irq_name = tep_find_field(context.irq, "name");
	memset(&context.max, 0, sizeof(context.max));
	trace_seq_init(&context.seq);
```

Simple tracefs example (irq-lat.c)

```
        tracefs_trace_off(NULL);
        tracefs_function_filter(NULL, "*common_interrupt*", NULL, TRACEFS_FL_RESET);
        tracefs_tracer_set(NULL, TRACEFS_TRACER_FUNCTION_GRAPH);
        tracefs_event_disable(NULL, NULL, NULL);
        tracefs_event_enable(NULL, "irq", "irq_handler_entry");
        tracefs_trace_on(NULL);
        sleep(5);
        tracefs_trace_off(NULL);

        tracefs_iterate_raw_events(context.tep, NULL, NULL, 0, callback, &context);

        if (context.max.start)
                printf("Max irq latency was %lld us from irq %s:%lld on cpu %d\n",
                        context.max.start / 1000, context.max.irq_name,
                         context.max.irq_vec, context.max_cpu);
        else
                printf("No interrupt was recorded ;-(\n");
        tracefs_tracer_clear(NULL);
        tracefs_function_filter(NULL, NULL, NULL, TRACEFS_FL_RESET);
        tracefs_event_disable(NULL, NULL, NULL);
        return 0;
}
```

Simple tracefs example (irq-lat.c)

```c
static int callback(struct tep_event *event, struct tep_record *record,
                     int cpu, void *data)
{
        unsigned long long delta;
        struct irq_context *context = data;
        int type = tep_data_type(context->tep, record);

        if (cpu >= context->nr_cpus)
                return 0;

        if (type == context->irq->id) {
                if (context->cpus[cpu].start) {
                        trace_seq_reset(&context->seq);
                        tep_print_field_content(&context->seq, record->data, record->size, context->irq_name);
                        free(context->cpus[cpu].irq_name);
                        trace_seq_terminate(&context->seq);
                        context->cpus[cpu].irq_name = strdup(context->seq.buffer);
                        tep_read_number_field(context->irq_vec, record->data, &context->cpus[cpu].irq_vec);
                }
        } else if (type == context->func_enter->id) {
                context->cpus[cpu].start = record->ts;

        } else if (type == context->func_exit->id) {
                if (context->cpus[cpu].start) {
                        delta = record->ts - context->cpus[cpu].start;
                        if (delta > context->max.start) {
                                context->max.start = delta;
                                context->max.irq_vec = context->cpus[cpu].irq_vec;
                                context->max.irq_name = context->cpus[cpu].irq_name;
                                context->cpus[cpu].irq_name = NULL;
                                context->max_cpu = cpu;
                        }
                        context->cpus[cpu].start = 0;
                }
        }

        return 0;
}
```

Simple tracefs example (irq-lat.c)

```c
static int callback(struct tep_event *event, struct tep_record *record,
                    int cpu, void *data)
{
        unsigned long long delta;
        struct irq_context *context = data;
        int type = tep_data_type(context->tep, record);

        if (cpu >= context->nr_cpus)
                return 0;

        if (type == context->irq->id) {
                if (context->cpus[cpu].start) {
                        trace_seq_reset(&context->seq);
                        tep_print_field_content(&context->seq, record->data, record->size, context->irq_name);
                        free(context->cpus[cpu].irq_name);
                        trace_seq_terminate(&context->seq);
                        context->cpus[cpu].irq_name = strdup(context->seq.buffer);
                        tep_read_number_field(context->irq_vec, record->data, &context->cpus[cpu].irq_vec);
                }
        } else if (type == context->func_enter->id) {
                context->cpus[cpu].start = record->ts;

        } else if (type == context->func_exit->id) {
                if (context->cpus[cpu].start) {
                        delta = record->ts - context->cpus[cpu].start;
                        if (delta > context->max.start) {
                                context->max.start = delta;
                                context->max.irq_vec = context->cpus[cpu].irq_vec;
                                context->max.irq_name = context->cpus[cpu].irq_name;
                                context->cpus[cpu].irq_name = NULL;
                                context->max_cpu = cpu;
                        }
                        context->cpus[cpu].start = 0;
                }
        }

        return 0;
}
```

Simple tracefs example (irq-lat.c)

```
static int callback(struct tep_event *event, struct tep_record *record,
                     int cpu, void *data)
{
        unsigned long long delta;
        struct irq_context *context = data;
        int type = tep_data_type(context->tep, record);

        if (cpu >= context->nr_cpus)
                return 0;

        if (type == context->irq->id) {
                if (context->cpus[cpu].start) {
                        trace_seq_reset(&context->seq);
                        tep_print_field_content(&context->seq, record->data, record->size, context->irq_name);
                        free(context->cpus[cpu].irq_name);
                        trace_seq_terminate(&context->seq);
                        context->cpus[cpu].irq_name = strdup(context->seq.buffer);
                        tep_read_number_field(context->irq_vec, record->data, &context->cpus[cpu].irq_vec);
                }
        } else if (type == context->func_enter->id) {
                context->cpus[cpu].start = record->ts;

        } else if (type == context->func_exit->id) {
                if (context->cpus[cpu].start) {
                        delta = record->ts - context->cpus[cpu].start;
                        if (delta > context->max.start) {
                                context->max.start = delta;
                                context->max.irq_vec = context->cpus[cpu].irq_vec;
                                context->max.irq_name = context->cpus[cpu].irq_name;
                                context->cpus[cpu].irq_name = NULL;
                                context->max_cpu = cpu;
                        }
                        context->cpus[cpu].start = 0;
                }
        }

        return 0;
}
```

Simple tracefs example (irq-lat.c)

```c
static int callback(struct tep_event *event, struct tep_record *record,
                    int cpu, void *data)
{
        unsigned long long delta;
        struct irq_context *context = data;
        int type = tep_data_type(context->tep, record);

        if (cpu >= context->nr_cpus)
                return 0;

        if (type == context->irq->id) {
                if (context->cpus[cpu].start) {
                        trace_seq_reset(&context->seq);
                        tep_print_field_content(&context->seq, record->data, record->size, context->irq_name);
                        free(context->cpus[cpu].irq_name);
                        trace_seq_terminate(&context->seq);
                        context->cpus[cpu].irq_name = strdup(context->seq.buffer);
                        tep_read_number_field(context->irq_vec, record->data, &context->cpus[cpu].irq_vec);
                }
        } else if (type == context->func_enter->id) {
                context->cpus[cpu].start = record->ts;

        } else if (type == context->func_exit->id) {
                if (context->cpus[cpu].start) {
                        delta = record->ts - context->cpus[cpu].start;
                        if (delta > context->max.start) {
                                context->max.start = delta;
                                context->max.irq_vec = context->cpus[cpu].irq_vec;
                                context->max.irq_name = context->cpus[cpu].irq_name;
                                context->cpus[cpu].irq_name = NULL;
                                context->max_cpu = cpu;
                        }
                        context->cpus[cpu].start = 0;
                }
        }

        return 0;
}
```

Simple tracefs example (irq-lat.c)

```
static int callback(struct tep_event *event, struct tep_record *record,
                    int cpu, void *data)
{
        unsigned long long delta;
        struct irq_context *context = data;
        int type = tep_data_type(context->tep, record);

        if (cpu >= context->nr_cpus)
                return 0;

        if (type == context->irq->id) {
                if (context->cpus[cpu].start) {
                        trace_seq_reset(&context->seq);
                        tep_print_field_content(&context->seq, record->data, record->size, context->irq_name);
                        free(context->cpus[cpu].irq_name);
                        trace_seq_terminate(&context->seq);
                        context->cpus[cpu].irq_name = strdup(context->seq.buffer);
                        tep_read_number_field(context->irq_vec, record->data, &context->cpus[cpu].irq_vec);
                }
        } else if (type == context->func_enter->id) {
                context->cpus[cpu].start = record->ts;

        } else if (type == context->func_exit->id) {
                if (context->cpus[cpu].start) {
                        delta = record->ts - context->cpus[cpu].start;
                        if (delta > context->max.start) {
                                context->max.start = delta;
                                context->max.irq_vec = context->cpus[cpu].irq_vec;
                                context->max.irq_name = context->cpus[cpu].irq_name;
                                context->cpus[cpu].irq_name = NULL;
                                context->max_cpu = cpu;
                        }
                        context->cpus[cpu].start = 0;
                }
        }

        return 0;
}
```

Interrupt latency

```
# gcc -o irq-lat -g -Wall irq-lat.c  `pkg-config --cflags --libs libtracefs`
# ./irq-lat
Max irq latency was 19 us from irq ahci[0000:00:1f.2]:24 on cpu 0
```

Writing into the kernel buffer from the application

- **tracefs_print_init**()
  - Pre-initalize the tracefs printing to not need to do that during hot paths of the application

Writing into the kernel buffer from the application

- **tracefs_print_init**()
  - Pre-initalize the tracefs printing to not need to do that during hot paths of the application
- **tracefs_printf**()
  - Writes into the ring buffer (will call tracefs_print_init() if it is not already initialized)

Writing into the kernel buffer from the application

- **tracefs_print_init**()
  - Pre-initalize the tracefs printing to not need to do that during hot paths of the application
- **tracefs_printf**()
  - Writes into the ring buffer (will call tracefs_print_init() if it is not already initialized)
- **tracefs_vprintf**()
  - Same as tracefs_printf() but for va_list arguments

Writing into the kernel buffer from the application

- **tracefs_print_init**()
  - Pre-initalize the tracefs printing to not need to do that during hot paths of the application
- **tracefs_printf**()
  - Writes into the ring buffer (will call tracefs_print_init() if it is not already initialized)
- **tracefs_vprintf**()
  - Same as tracefs_printf() but for va_list arguments
- **tracefs_printf_close**()
  - Cleans up the open file descriptors from tracefs_print_init()

(Intermission) Let’s talk about unbounded priority inversion

- Common real-time problem

(Intermission) Let’s talk about unbounded priority inversion

- Common real-time problem
- 3 tasks
  - A - has the highest priority
  - B - has a priority in the middle
  - C - has the lowest priority

(Intermission) Let’s talk about unbounded priority inversion

- Common real-time problem
- 3 tasks
  - A - has the highest priority
  - B - has a priority in the middle
  - C - has the lowest priority
- C has a lock

(Intermission) Let’s talk about unbounded priority inversion

- Common real-time problem
- 3 tasks
  - A - has the highest priority
  - B - has a priority in the middle
  - C - has the lowest priority
- C has a lock
- A tries to take it and blocks (waiting on C)

(Intermission) Let’s talk about unbounded priority inversion

- Common real-time problem
- 3 tasks
  - A - has the highest priority
  - B - has a priority in the middle
  - C - has the lowest priority
- C has a lock
- A tries to take it and blocks (waiting on C)
- B wakes up, preempts C and keeps C from continuing

(Intermission) Let’s talk about unbounded priority inversion

- Common real-time problem
- 3 tasks
  - A - has the highest priority
  - B - has a priority in the middle
  - C - has the lowest priority
- C has a lock
- A tries to take it and blocks (waiting on C)
- B wakes up, preempts C and keeps C from continuing
- C can not release the lock and thus B keeps A from running

(Intermission) Let’s talk about unbounded priority inversion

- Common real-time problem
- 3 tasks
  - A - has the highest priority
  - B - has a priority in the middle
  - C - has the lowest priority
- C has a lock
- A tries to take it and blocks (waiting on C)
- B wakes up, preempts C and keeps C from continuing
- C can not release the lock and thus B keeps A from running
- Solved with Priority Inheritance

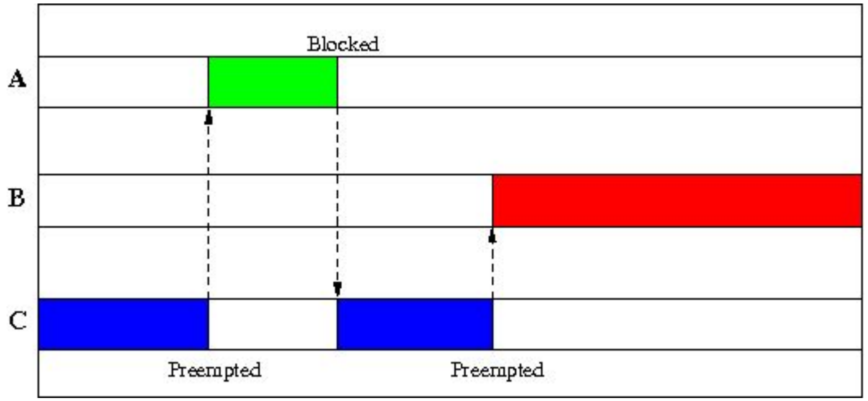
(Intermission) Let’s talk about unbounded priority inversion

- Common real-time problem
- 3 tasks
  - A - has the highest priority
  - B - has a priority in the middle
  - C - has the lowest priority
- C has a lock
- A tries to take it and blocks (waiting on C)
- B wakes up, preempts C and keeps C from continuing
- C can not release the lock and thus B keeps A from running
- Solved with Priority Inheritance
- C inherits A’s priority when A blocks on a lock owned by C

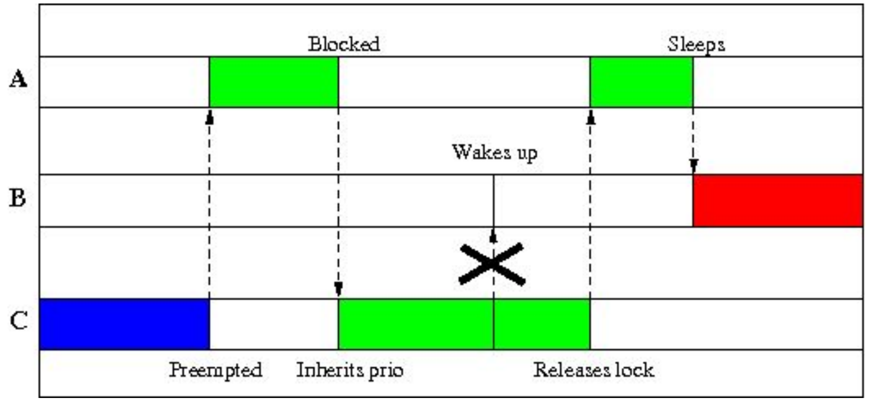
(Intermission) Let’s talk about unbounded priority inversion

- Common real-time problem
- 3 tasks
  - A - has the highest priority
  - B - has a priority in the middle
  - C - has the lowest priority
- C has a lock
- A tries to take it and blocks (waiting on C)
- B wakes up, preempts C and keeps C from continuing
- C can not release the lock and thus B keeps A from running
- Solved with Priority Inheritance
- C inherits A’s priority when A blocks on a lock owned by C
- B can no longer preempt C

Unbounded Priority Inversion



Bounded Priority Inversion (using Priority Inheritance)



Priority inheritance example (pi-test.c)

```
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <unistd.h>
#define __USE_GNU
#include <sys/syscall.h>
#include <pthread.h>
#include <sched.h>
#include <tracefs.h>

static void trace(const char *fmt, ...)
{
        va_list ap, ap2;

        va_start(ap, fmt);
        va_copy(ap, ap2);
        tracefs_vprintf(NULL, fmt, ap);
        vprintf(fmt, ap2);
        va_end(ap2);
        va_end(ap);
}
```

Priority inheritance example (pi-test.c)

```
#define MAIN_PRIO 6
#define A_PRIO 5
#define B_PRIO 4
#define C_PRIO 2
#define SLEEP_SECS 5

pthread_mutex_t L_lock;

static pthread_barrier_t start_A;
static pthread_barrier_t start_B;
static pthread_barrier_t start_C;
static bool B_spins_a_long_time = true;
static bool B_ran_a_lot;
static bool PI_has_failed;

#define barrier() asm volatile (" " : : : "memory")
#define gettid() syscall(__NR_gettid)

static void set_prio(int prio)
{
        struct sched_param sp = { .sched_priority = prio };
        sched_setscheduler(gettid(), SCHED_FIFO, &sp);
}
```

Priority inheritance example (pi-test.c)

```
static void *thread_A(void *arg)
{
        set_prio(A_PRIO);

        pthread_barrier_wait(&start_A);
        trace("A started\n");

        trace("A waking up B\n");
        pthread_barrier_wait(&start_B);

        trace("A grabbing lock\n");
        pthread_mutex_lock(&L_lock);
        if (B_ran_a_lot)
                PI_has_failed = true;
        trace("A has lock\n");
        pthread_mutex_unlock(&L_lock);
        trace("A released lock\n");

        trace("A exits\n");
        return NULL;
}
```

Priority inheritance example (pi-test.c)

```
static void *thread_B(void *arg)
{
        set_prio(B_PRIO);

        pthread_barrier_wait(&start_B);

        trace("B started\n");

        while (B_spins_a_long_time)
                barrier();

        B_ran_a_lot = 1;

        trace("B exits\n");
        return NULL;
}
```

Priority inheritance example (pi-test.c)

```
static void *thread_C(void *arg)
{
        unsigned long long i;

        set_prio(C_PRIO);

        pthread_barrier_wait(&start_C);
        trace("C started\n");

        pthread_mutex_lock(&L_lock);
        trace("C has lock\n");

        pthread_barrier_wait(&start_A);

        for (i=0; i < 1000000000; i++)
                barrier();

        if (B_ran_a_lot)
                trace("C ran after B\n");

        trace("C releasing lock\n");
        pthread_mutex_unlock(&L_lock);
        trace("C no longer has lock\n");

        trace("C exits\n");
        return NULL;
}
```

Priority inheritance example (pi-test.c)

```
int main (int argc, char **argv)
{
        pthread_mutexattr_t attr;
        cpu_set_t cpumask;
        pthread_t A,B,C;
        int secs = SLEEP_SECS;

        CPU_ZERO(&cpumask);
        CPU_SET(0, &cpumask);
        sched_setaffinity(0, sizeof(cpumask), &cpumask);

        tracefs_print_init(NULL);

        tracefs_event_disable(NULL, NULL, NULL);
        tracefs_tracer_clear(NULL);
        tracefs_instance_file_write(NULL, "trace", "");
        tracefs_event_enable(NULL, "sched", "sched_waking");
        tracefs_event_enable(NULL, "sched", "sched_switch");
        tracefs_event_enable(NULL, "sched", "sched_pi_setprio");
```

Priority inheritance example (pi-test.c)

```
pthread_mutexattr_init(&attr);
pthread_mutex_init(&L_lock, &attr);

pthread_barrier_init(&start_A, NULL, 2);
pthread_barrier_init(&start_C, NULL, 2);
pthread_barrier_init(&start_B, NULL, 2);

pthread_create(&A, NULL, thread_A, NULL);
pthread_create(&B, NULL, thread_B, NULL);
pthread_create(&C, NULL, thread_C, NULL);

set_prio(MAIN_PRIO);

tracefs_trace_on(NULL);
trace("Let er rip!\n");
pthread_barrier_wait(&start_C);

sleep(secs);

trace("Stopping B\n");
B_spins_a_long_time = false;
```

Priority inheritance example (pi-test.c)

```
	trace("wait for A\n");
	pthread_join(A, NULL);
	trace("wait for B\n");
	pthread_join(B, NULL);
	trace("wait for C\n");
	pthread_join(C, NULL);

	tracefs_trace_off(NULL);
	tracefs_print_close(NULL);

	if (PI_has_failed)
		printf("Priority inheritance failed\n");
	else
		printf("Priority inheritance worked like a charm!\n");

	exit(0);
}
```

Priority inheritance tracing

```
# gcc -o pi-test -g -Wall pi-test.c  `pkg-config --cflags --libs libtracefs` -lpthread
# ./pi-test
Let er rip!
C started
C has lock
A started
A waking up B
A grabbing lock
B started
Stopping B
wait for A
B exits
C ran after B
C releasing lock
A has lock
A released lock
A exits
wait for B
wait for B
C no longer has lock
C exits
Priority inheritance failed
```

Priority inheritance tracing

```
# trace-cmd show
[..]
#           TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#              | |         |   |||||     |         |
         pi-test-26594   [000] ..... 203652.575514: tracing_mark_write: Let er rip!
         pi-test-26594   [000] d..3. 203652.575524: sched_waking: comm=kworker/u16:2 pid=26526 prio=120 target_cpu=000
         pi-test-26594   [000] d..2. 203652.575542: sched_switch: prev_comm=pi-test prev_pid=26594 prev_prio=93 prev_state=
          <idle>-0       [007] d..2. 203652.575549: sched_switch: prev_comm=swapper/7 prev_pid=0 prev_prio=120 prev_state=R
         pi-test-26597   [000] d..2. 203652.575549: sched_waking: comm=pi-test pid=26594 prio=93 target_cpu=000
         pi-test-26597   [000] d..2. 203652.575551: sched_switch: prev_comm=pi-test prev_pid=26597 prev_prio=97 prev_state=
   kworker/u16:2-26526   [007] d..3. 203652.575552: sched_waking: comm=sshd pid=15865 prio=120 target_cpu=004
         pi-test-26594   [000] d..2. 203652.575554: sched_switch: prev_comm=pi-test prev_pid=26594 prev_prio=93 prev_state=
          <idle>-0       [004] d..2. 203652.575559: sched_switch: prev_comm=swapper/4 prev_pid=0 prev_prio=120 prev_state=R
   kworker/u16:2-26526   [007] d..2. 203652.575561: sched_switch: prev_comm=kworker/u16:2 prev_pid=26526 prev_prio=120
prev_state=I ==> next_comm=swapper/7 next_pid=0 next_prio=120
         pi-test-26597   [000] ..... 203652.575586: tracing_mark_write: C started
         pi-test-26597   [000] d..3. 203652.575590: sched_waking: comm=kworker/u16:2 pid=26526 prio=120 target_cpu=007
          <idle>-0       [007] d..2. 203652.575593: sched_switch: prev_comm=swapper/7 prev_pid=0 prev_prio=120 prev_state=R
   kworker/u16:2-26526   [007] d..2. 203652.575596: sched_switch: prev_comm=kworker/u16:2 prev_pid=26526 prev_prio=120
prev_state=I ==> next_comm=swapper/7 next_pid=0 next_prio=120
         pi-test-26597   [000] ..... 203652.575597: tracing_mark_write: C has lock
         pi-test-26597   [000] d..3. 203652.575602: sched_waking: comm=kworker/u16:2 pid=26526 prio=120 target_cpu=007
          <idle>-0       [007] d..2. 203652.575604: sched_switch: prev_comm=swapper/7 prev_pid=0 prev_prio=120 prev_state=R
   kworker/u16:2-26526   [007] d..2. 203652.575606: sched_switch: prev_comm=kworker/u16:2 prev_pid=26526 prev_prio=120 prev
         pi-test-26597   [000] d..2. 203652.575608: sched_switch: prev_comm=pi-test prev_pid=26597 prev_prio=97 prev_state=
         pi-test-26596   [000] d..2. 203652.575619: sched_switch: prev_comm=pi-test prev_pid=26596 prev_prio=95 prev_state=
         pi-test-26595   [000] d..2. 203652.575634: sched_waking: comm=pi-test pid=26597 prio=97 target_cpu=000
[..]
```

Priority inheritance tracing

```
# trace-cmd show
[..]
#            TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#               | |         |   |||||     |         |
         pi-test-26594   [000] ..... 203652.575514: tracing_mark_write: Let er rip!
         pi-test-26594   [000] d..3. 203652.575524: sched_waking: comm=kworker/u16:2 pid=26526 prio=120 target_cpu=000
         pi-test-26594   [000] d..2. 203652.575542: sched_switch: prev_comm=pi-test prev_pid=26594 prev_prio=93 prev_state=
          <idle>-0       [007] d..2. 203652.575549: sched_switch: prev_comm=swapper/7 prev_pid=0 prev_prio=120 prev_state=R
         pi-test-26597   [000] d..2. 203652.575549: sched_waking: comm=pi-test pid=26594 prio=93 target_cpu=000
         pi-test-26597   [000] d..2. 203652.575551: sched_switch: prev_comm=pi-test prev_pid=26597 prev_prio=97 prev_state=
   kworker/u16:2-26526   [007] d..3. 203652.575552: sched_waking: comm=sshd pid=15865 prio=120 target_cpu=004
         pi-test-26594   [000] d..2. 203652.575554: sched_switch: prev_comm=pi-test prev_pid=26594 prev_prio=93 prev_state=
          <idle>-0       [004] d..2. 203652.575559: sched_switch: prev_comm=swapper/4 prev_pid=0 prev_prio=120 prev_state=R
   kworker/u16:2-26526   [007] d..2. 203652[illegible]556[illegible]: sched_switch: prev_comm=kworker/u16:2 prev_pid=26526 prev_prio=120
prev_state=I ==> next_comm=swapper/7 next_pid=0 next_prio=12[illegible]
         pi-test-26597   [000] ..... 203652[illegible]: tracing_mark_write: C started
         pi-test-26597   [000] d..3. 203652[illegible]: sched_waking: comm=kworker/u16:2 pid=26526 prio=120 target_cpu=007
          <idle>-0       [007] d..2. 203652.575593: sched_switch: prev_comm=swapper/7 prev_pid=0 prev_prio=120 prev_state=R
   kworker/u16:2-26526   [007] d..2. 203652.575596: sched_switch: prev_comm=kworker/u16:2 prev_pid=26526 prev_prio=120
prev_state=I ==> next_comm=swapper/7 next_pid=0 next_prio=120
         pi-test-26597   [000] ..... 203652.575597: tracing_mark_write: C has lock
         pi-test-26597   [000] d..3. 203[illegible].57[illegible]602: sched_waking: comm=kworker/u16:2 pid=26526 prio=120 target_cpu=007
          <idle>-0       [007] d..2. 203[illegible]: sched_switch: prev_comm=swapper/7 prev_pid=0 prev_prio=120 prev_state=R
   kworker/u16:2-26526   [007] d..2. 203[illegible]: sched_switch: prev_comm=kworker/u16:2 prev_pid=26526 prev_prio=120 prev
         pi-test-26597   [000] d..2. 203[illegible]52.[illegible]608: sched_switch: prev_comm=pi-test prev_pid=26597 prev_prio=97 prev_state=
         pi-test-26596   [000] d..2. 203652.575619: sched_switch: prev_comm=pi-test prev_pid=26596 prev_prio=95 prev_state=
         pi-test-26595   [000] d..2. 203652.575634: sched_waking: comm=pi-test pid=26597 prio=97 target_cpu=000
[..]
```

TOO MUCH NOISE!

Priority inheritance example (pi-test.c)

```
int main (int argc, char **argv)
{
        pthread_mutexattr_t attr;
        cpu_set_t cpumask;
        pthread_t A,B,C;
        int secs = SLEEP_SECS;
        char pid[24];

        CPU_ZERO(&cpumask);
        CPU_SET(0, &cpumask);
        sched_setaffinity(0, sizeof(cpumask), &cpumask);

        sprintf(pid, "%ld", gettid());

        tracefs_print_init(NULL);

        tracefs_event_disable(NULL, NULL, NULL);
        tracefs_tracer_clear(NULL);
        tracefs_instance_file_write(NULL, "trace", "");
        tracefs_event_enable(NULL, "sched", "sched_waking");
        tracefs_event_enable(NULL, "sched", "sched_switch");
        tracefs_event_enable(NULL, "sched", "sched_pi_setprio");
        tracefs_instance_file_write(NULL, "set_event_pid", pid);
```

Priority inheritance tracing

```
# gcc -o pi-test -g -Wall pi-test.c  `pkg-config --cflags --libs libtracefs` -lpthread
# ./pi-test
Let er rip!
C started
C has lock
A started
A waking up B
A grabbing lock
B started
Stopping B
wait for A
B exits
C ran after B
C releasing lock
A has lock
A released lock
A exits
wait for B
wait for B
C no longer has lock
C exits
Priority inheritance failed
```

Priority inheritance tracing

```
# trace-cmd show
[..]
#           TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#              | |         |   |||||     |         |
         pi-test-26780   [000] ..... 204476.898726: tracing_mark_write: Let er rip!
         pi-test-26780   [000] d..3. 204476.898731: sched_waking: comm=kworker/u16:3 pid=26619 prio=120 target_cpu=000
         pi-test-26780   [000] d..2. 204476.898741: sched_switch: prev_comm=pi-test prev_pid=26780 prev_prio=93 prev_state=S ==> next_comm=pi
         pi-test-26783   [000] d..2. 204476.898748: sched_waking: comm=pi-test pid=26780 prio=93 target_cpu=000
         pi-test-26783   [000] d..2. 204476.898750: sched_switch: prev_comm=pi-test prev_pid=26783 prev_prio=97 prev_state=R+ ==> next_comm=p
         pi-test-26780   [000] d..2. 204476.898754: sched_switch: prev_comm=pi-test prev_pid=26780 prev_prio=93 prev_state=S ==> next_comm=pi
         pi-test-26783   [000] ..... 204476.898784: tracing_mark_write: C started
         pi-test-26783   [000] ..... 204476.898791: tracing_mark_write: C has lock
         pi-test-26781   [000] ..... 204476.898827: tracing_mark_write: A started
         pi-test-26781   [000] ..... 204476.898833: tracing_mark_write: A waking up B
         pi-test-26781   [000] ..... 204476.898838: tracing_mark_write: A grabbing lock
         pi-test-26782   [000] ..... 204476.898853: tracing_mark_write: B started
         pi-test-26782   [000] d.h2. 204481.898719: sched_waking: comm=pi-test pid=26780 prio=93 target_cpu=000
         pi-test-26782   [000] d..2. 204481.898737: sched_switch: prev_comm=pi-test prev_pid=26782 prev_prio=95 prev_state=R ==> next_comm=pi
         pi-test-26780   [000] ..... 204481.898749: tracing_mark_write: Stopping B
         pi-test-26780   [000] d..3. 204481.898760: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=007
         pi-test-26780   [000] ..... 204481.898776: tracing_mark_write: wait for A
         pi-test-26780   [000] d..2. 204481.898792: sched_switch: prev_comm=pi-test prev_pid=26780 prev_prio=93 prev_state=S ==> next_comm=pi
         pi-test-26782   [000] ..... 204481.898794: tracing_mark_write: B exits
         pi-test-26783   [000] ..... 204483.498843: tracing_mark_write: C ran after B
         pi-test-26783   [000] ..... 204483.498851: tracing_mark_write: C releasing lock
         pi-test-26781   [000] ..... 204483.498860: tracing_mark_write: A has lock
         pi-test-26781   [000] ..... 204483.498863: tracing_mark_write: A released lock
         pi-test-26781   [000] ..... 204483.498866: tracing_mark_write: A exits
         pi-test-26781   [000] d..2. 204483.498875: sched_waking: comm=pi-test pid=26780 prio=93 target_cpu=000
         pi-test-26781   [000] d..2. 204483.498877: sched_switch: prev_comm=pi-test prev_pid=26781 prev_prio=94 prev_state=R+ ==> next_comm=p
         pi-test-26780   [000] ..... 204483.498879: tracing_mark_write: wait for B
         pi-test-26780   [000] d..3. 204483.498881: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=007
         pi-test-26780   [000] ..... 204483.498892: tracing_mark_write: wait for C
         pi-test-26780   [000] d..2. 204483.498896: sched_switch: prev_comm=pi-test prev_pid=26780 prev_prio=93 prev_state=S ==> next_comm=pi
         pi-test-26783   [000] ..... 204483.498908: tracing_mark_write: C no longer has lock
         pi-test-26783   [000] ..... 204483.498913: tracing_mark_write: C exits
         pi-test-26783   [000] d..2. 204483.498919: sched_waking: comm=pi-test pid=26780 prio=93 target_cpu=000
         pi-test-26783   [000] d..2. 204483.498921: sched_switch: prev_comm=pi-test prev_pid=26783 prev_prio=97 prev_state=R+ ==> next_comm=p
```

Priority inheritance tracing

```
# trace-cmd show
[..]
#           TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#              | |         |   |||||     |         |
         pi-test-26780   [000] ..... 204476.898726: tracing_mark_write: Let er rip!
         pi-test-26780   [000] d..3. 204476.898731: sched_waking: comm=kworker/u16:3 pid=26619 prio=120 target_cpu=000
         pi-test-26780   [000] d..2. 204476.898741: sched_switch: prev_comm=pi-test prev_pid=26780 prev_prio=93 prev_state=S ==> next_comm=pi
         pi-test-26783   [000] d..2. 204476.898748: sched_waking: comm=pi-test pid=26780 prio=93 target_cpu=000
         pi-test-26783   [000] d..2. 204476.898750: sched_switch: prev_comm=pi-test prev_pid=26783 prev_prio=97 prev_state=R+ ==> next_comm=p
         pi-test-26780   [000] d..2. 204476.898754: sched_switch: prev_comm=pi-test prev_pid=26780 prev_prio=93 prev_state=S ==> next_comm=pi
         pi-test-26783   [000] ..... 204476.898784: tracing_mark_write: C started
         pi-test-26783   [000] ..... 204476.898791: tracing_mark_write: C has lock
         pi-test-26781   [000] ..... 204476.898827: tracing_mark_write: A started
         pi-test-26781   [000] ..... 204476.898833: tracing_mark_write: A waking up B
         pi-test-26781   [000] ..... 204476.898838: tracing_mark_write: A grabbing lock
         pi-test-26782   [000] ..... 204476.898853: tracing_mark_write: B started
         pi-test-26782   [000] d.h2. 204481.898719: sched_waking: comm=pi-test pid=26780 prio=93 target_cpu=000
         pi-test-26782   [000] d..2. 204481.898737: sched_switch: prev_comm=pi-test prev_pid=26782 prev_prio=95 prev_state=R ==> next_comm=pi
         pi-test-26780   [000] ..... 204481.898749: tracing_mark_write: Stopping B
         pi-test-26780   [000] d..3. 204481.898760: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=007
         pi-test-26780   [000] ..... 204481.898776: tracing_mark_write: wait for A
         pi-test-26780   [000] d..2. 204481.898792: sched_switch: prev_comm=pi-test prev_pid=26780 prev_prio=93 prev_state=S ==> next_comm=pi
         pi-test-26782   [000] ..... 204481.898794: tracing_mark_write: B exits
         pi-test-26783   [000] ..... 204483.498843: tracing_mark_write: C ran after B
         pi-test-26783   [000] ..... 204483.498851: tracing_mark_write: C releasing lock
         pi-test-26781   [000] ..... 204483.498860: tracing_mark_write: A has lock
         pi-test-26781   [000] ..... 204483.498863: tracing_mark_write: A released lock
         pi-test-26781   [000] ..... 204483.498866: tracing_mark_write: A exits
         pi-test-26781   [000] d..2. 204483.498875: sched_waking: comm=pi-test pid=26780 prio=93 target_cpu=000
         pi-test-26781   [000] d..2. 204483.498877: sched_switch: prev_comm=pi-test prev_pid=26781 prev_prio=94 prev_state=R+ ==> next_comm=p
         pi-test-26780   [000] ..... 204483.498879: tracing_mark_write: wait for B
         pi-test-26780   [000] d..3. 204483.498881: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=007
         pi-test-26780   [000] ..... 204483.498892: tracing_mark_write: wait for C
         pi-test-26780   [000] d..2. 204483.498896:                                 pid=26780 prev_prio=93 prev_state=S ==> next_comm=pi
         pi-test-26783   [000] ..... 204483.498908:                                 ock
         pi-test-26783   [000] ..... 204483.498913:
         pi-test-26783   [000] d..2. 204483.498919: sched_waking: comm=pi-test pid=26780 prio=93 target_cpu=000
         pi-test-26783   [000] d..2. 204483.498921: sched_switch: prev_comm=pi-test prev_pid=26783 prev_prio=97 prev_state=R+ ==> next_comm=p
```

No data for the threads

Priority inheritance example (pi-test.c)

```
int main (int argc, char **argv)
{
        pthread_mutexattr_t attr;
        cpu_set_t cpumask;
        pthread_t A,B,C;
        int secs = SLEEP_SECS;
        char pid[24];

        CPU_ZERO(&cpumask);
        CPU_SET(0, &cpumask);
        sched_setaffinity(0, sizeof(cpumask), &cpumask);

        sprintf(pid, "%ld", gettid());

        tracefs_print_init(NULL);

        tracefs_event_disable(NULL, NULL, NULL);
        tracefs_tracer_clear(NULL);
        tracefs_instance_file_write(NULL, "trace", "");
        tracefs_event_enable(NULL, "sched", "sched_waking");
        tracefs_event_enable(NULL, "sched", "sched_switch");
        tracefs_event_enable(NULL, "sched", "sched_pi_setprio");
        tracefs_instance_file_write(NULL, "set_event_pid", pid);
        tracefs_option_enable(NULL, TRACEFS_OPTION_EVENT_FORK);
```

Priority inheritance tracing

```
# gcc -o pi-test -g -Wall pi-test.c  `pkg-config --cflags --libs libtracefs` -lpthread
# ./pi-test
Let er rip!
C started
C has lock
A started
A waking up B
A grabbing lock
B started
Stopping B
wait for A
B exits
C ran after B
C releasing lock
A has lock
A released lock
A exits
wait for B
wait for B
C no longer has lock
C exits
Priority inheritance failed
```

Priority inheritance tracing

```
# trace-cmd show
[..]
#           TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#              | |         |   |||||     |         |
         pi-test-26874   [000] ..... 205331.438350: tracing_mark_write: Let er rip!
         pi-test-26874   [000] d..3. 205331.438356: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=000
         pi-test-26874   [000] d..2. 205331.438366: sched_switch: prev_comm=pi-test prev_pid=26874 prev_prio=93 prev_state=S ==> next_comm=pi
         pi-test-26877   [000] d..2. 205331.438373: sched_waking: comm=pi-test pid=26874 prio=93 target_cpu=000
         pi-test-26877   [000] d..2. 205331.438375: sched_switch: prev_comm=pi-test prev_pid=26877 prev_prio=97 prev_state=R+ ==> next_comm=p
         pi-test-26874   [000] d..2. 205331.438379: sched_switch: prev_comm=pi-test prev_pid=26874 prev_prio=93 prev_state=S ==> next_comm=pi
         pi-test-26877   [000] ..... 205331.438401: tracing_mark_write: C started
         pi-test-26877   [000] d..3. 205331.438404: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=002
         pi-test-26877   [000] ..... 205331.438409: tracing_mark_write: C has lock
         pi-test-26877   [000] d..3. 205331.438411: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=002
         pi-test-26877   [000] d..2. 205331.438416: sched_switch: prev_comm=pi-test prev_pid=26877 prev_prio=97 prev_state=S ==> next_comm=pi
         pi-test-26876   [000] d..2. 205331.438423: sched_switch: prev_comm=pi-test prev_pid=26876 prev_prio=95 prev_state=S ==> next_comm=pi
         pi-test-26875   [000] d..2. 205331.438428: sched_waking: comm=pi-test pid=26877 prio=97 target_cpu=000
         pi-test-26875   [000] ..... 205331.438444: tracing_mark_write: A started
         pi-test-26875   [000] d..3. 205331.438447: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=002
         pi-test-26875   [000] ..... 205331.438450: tracing_mark_write: A waking up B
         pi-test-26875   [000] d..3. 205331.438451: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=002
         pi-test-26875   [000] d..2. 205331.438454: sched_waking: comm=pi-test pid=26876 prio=95 target_cpu=000
         pi-test-26875   [000] ..... 205331.438464: tracing_mark_write: A grabbing lock
         pi-test-26875   [000] d..3. 205331.438466: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=002
         pi-test-26875   [000] d..2. 205331.438470: sched_switch: prev_comm=pi-test prev_pid=26875 prev_prio=94 prev_state=S ==> next_comm=pi
         pi-test-26876   [000] ..... 205331.438482: tracing_mark_write: B started
         pi-test-26876   [000] d..3. 205331.438484: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=002
         pi-test-26876   [000] d.s3. 205331.442296: sched_waking: comm=kworker/0:2 pid=25205 prio=120 target_cpu=000
         pi-test-26876   [000] d.s2. 205331.458295: sched_waking: comm=kcompactd0 pid=82 prio=120 target_cpu=003
         pi-test-26876   [000] d.s2. 205331.674294: sched_waking: comm=jbd2/dm-0-8 pid=558 prio=120 target_cpu=006
         pi-test-26876   [000] d.s3. 205331.674629: sched_waking: comm=jbd2/dm-0-8 pid=558 prio=120 target_cpu=006
[..]
```

Priority inheritance tracing

```
# trace-cmd show
[..]
#           TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#              | |         |   |||||     |         |
         pi-test-26874   [000] ..... 205331.438350: tracing_mark_write: Let er rip!
         pi-test-26874   [000] d..3. 205331.438356: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=000
         pi-test-26874   [000] d..2. 205331.438366: sched_switch: prev_comm=pi-test prev_pid=26874 prev_prio=93 prev_state=S ==> next_comm=pi
         pi-test-26877   [000] d..2. 205331.438373: sched_waking: comm=pi-test pid=26874 prio=93 target_cpu=000
         pi-test-26877   [000] d..2. 205331.438375: sched_switch: prev_comm=pi-test prev_pid=26877 prev_prio=97 prev_state=R+ ==> next_comm=p
         pi-test-26874   [000] d..2. 205331.438379: sched_switch: prev_comm=pi-test prev_pid=26874 prev_prio=93 prev_state=S ==> next_comm=pi
         pi-test-26877   [000] ..... 205331.438401: tracing_mark_write: C started
         pi-test-26877   [000] d..3. 205331.438404: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=002
         pi-test-26877   [000] ..... 205331.438409: tracing_mark_write: C has lock
         pi-test-26877   [000] d..3. 205331.438411: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=002
         pi-test-26877   [000] d..2. 205331.438416: sched_switch: prev_comm=pi-test prev_pid=26877 prev_prio=97 prev_state=S ==> next_comm=pi
         pi-test-26876   [000] d..2. 205331.438423: sched_switch: prev_comm=pi-test prev_pid=26876 prev_prio=95 prev_state=S ==> next_comm=pi
         pi-test-26875   [000] d..2. 205331.438428: sched_waking: comm=pi-test pid=26877 prio=97 target_cpu=000
         pi-test-26875   [000] ..... 205331.438444: tracing_mark_write: A started
         pi-test-26875   [000] d..3. 205331.438447: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=002
         pi-test-26875   [000] ..... 205331.438450: tracing_mark_write: A waking up
         pi-test-26875   [000] d..3. 205331.438451: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=002
         pi-test-26875   [000] d..2. 205331.438454: sched_waking: comm=pi-test pid=26876 prio=95 target_cpu=000
         pi-test-26875   [000] ..... 205331.438464: tracing_mark_write: A grabbing lock
         pi-test-26875   [000] d..3. 205331.438466: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=002
         pi-test-26875   [000] d..2. 205331.438470: sched_switch: prev_comm=pi-test prev_pid=26875 prev_prio=94 prev_state=S ==> next_comm=pi
         pi-test-26876   [000] ..... 205331.438482: tracing_mark_write: B started
         pi-test-26876   [000] d..3. 205331.438484: sched_waking: comm=kworker/u16:0 pid=26572 prio=120 target_cpu=002
         pi-test-26876   [000] d.s3. 205331.442296: sched_waking: comm=kworker/0:2 pid=25205 prio=120 target_cpu=000
         pi-test-26876   [000] d.s2. 205331.458295: sched_waking: comm=kcompactd0 pid=82 prio=120 target_cpu=003
         pi-test-26876   [000] d.s2. 205331.674294: sched_waking: comm=jbd2/dm-0-8 pid=558 prio=120 target_cpu=006
         pi-test-26876   [000] d.s3. 205331.674629: sched_waking: comm=jbd2/dm-0-8 pid=558 prio=120 target_cpu=006
[..]
```

MORE NOISE!

Priority inheritance example (pi-test.c)

```
int main (int argc, char **argv)
{
        pthread_mutexattr_t attr;
        struct tep_handle *tep;
        struct tep_event *sched_waking;
        cpu_set_t cpumask;
        pthread_t A,B,C;
        int secs = SLEEP_SECS;
        char pid[24];

        CPU_ZERO(&cpumask);
        CPU_SET(0, &cpumask);
        sched_setaffinity(0, sizeof(cpumask), &cpumask);

        sprintf(pid, "%ld", gettid());

        tracefs_print_init(NULL);

        tep = tracefs_local_events(NULL);
        sched_waking = tep_find_event_by_name(tep, "sched", "sched_waking");
        tracefs_event_filter_apply(NULL, sched_waking, "comm == \"pi-test\"");

        tracefs_event_disable(NULL, NULL, NULL);
        tracefs_tracer_clear(NULL);
        tracefs_instance_file_write(NULL, "trace", "");
        tracefs_event_enable(NULL, "sched", "sched_waking");
        tracefs_event_enable(NULL, "sched", "sched_switch");
```

Priority inheritance tracing

```
# trace-cmd show
[..]
#           TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#              | |         |   |||||     |         |
         pi-test-27042   [000] ..... 205864.784655: tracing_mark_write: Let er rip!
         pi-test-27042   [000] d..2. 205864.784680: sched_switch: prev_comm=pi-test prev_pid=27042 prev_prio=93 prev_state=S ==> next_comm=pi
         pi-test-27045   [000] d..2. 205864.784690: sched_waking: comm=pi-test pid=27042 prio=93 target_cpu=000
         pi-test-27045   [000] d..2. 205864.784694: sched_switch: prev_comm=pi-test prev_pid=27045 prev_prio=97 prev_state=R+ ==> next_comm=p
         pi-test-27042   [000] d..2. 205864.784699: sched_switch: prev_comm=pi-test prev_pid=27042 prev_prio=93 prev_state=S ==> next_comm=pi
         pi-test-27045   [000] ..... 205864.784722: tracing_mark_write: C started
         pi-test-27045   [000] ..... 205864.784731: tracing_mark_write: C has lock
         pi-test-27045   [000] d..2. 205864.784740: sched_switch: prev_comm=pi-test prev_pid=27045 prev_prio=97 prev_state=S ==> next_comm=pi
         pi-test-27044   [000] d..2. 205864.784749: sched_switch: prev_comm=pi-test prev_pid=27044 prev_prio=95 prev_state=S ==> next_comm=pi
         pi-test-27043   [000] d..2. 205864.784755: sched_waking: comm=pi-test pid=27045 prio=97 target_cpu=000
         pi-test-27043   [000] ..... 205864.784771: tracing_mark_write: A started
         pi-test-27043   [000] ..... 205864.784778: tracing_mark_write: A waking up B
         pi-test-27043   [000] d..2. 205864.784784: sched_waking: comm=pi-test pid=27044 prio=95 target_cpu=000
         pi-test-27043   [000] ..... 205864.784787: tracing_mark_write: A grabbing lock
         pi-test-27043   [000] d..2. 205864.784795: sched_switch: prev_comm=pi-test prev_pid=27043 prev_prio=94 prev_state=S ==> next_comm=pi
         pi-test-27044   [000] ..... 205864.784808: tracing_mark_write: B started
         pi-test-27044   [000] d..2. 205864.914286: sched_switch: prev_comm=pi-test prev_pid=27044 prev_prio=95 prev_state=R ==> next_comm=mi
     migration/0-19      [000] d..2. 205864.914290: sched_switch: prev_comm=migration/0 prev_pid=19 prev_prio=0 prev_state=S ==> next_comm=pi
         pi-test-27044   [000] d..2. 205865.736277: sched_switch: prev_comm=pi-test prev_pid=27044 prev_prio=95 prev_state=R ==> next_comm=kw
          <idle>-0       [000] d..2. 205865.784646: sched_switch: prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=R ==> next_comm=pi-
```

Priority inheritance example (pi-test.c)

```
pthread_mutexattr_init(&attr);
pthread_mutexattr_setprotocol(&attr, PTHREAD_PRIO_INHERIT);
pthread_mutex_init(&L_lock, &attr);

pthread_barrier_init(&start_A, NULL, 2);
pthread_barrier_init(&start_C, NULL, 2);
pthread_barrier_init(&start_B, NULL, 2);

pthread_create(&A, NULL, thread_A, NULL);
pthread_create(&B, NULL, thread_B, NULL);
pthread_create(&C, NULL, thread_C, NULL);

set_prio(MAIN_PRIO);

tracefs_trace_on(NULL);
trace("Let er rip!\n");
pthread_barrier_wait(&start_C);

sleep(secs);

trace("Stopping B\n");
B_spins_a_long_time = false;
```

Priority inheritance tracing

```
# gcc -o pi-test -g -Wall pi-test.c  `pkg-config --cflags --libs libtracefs` -lpthread
# ./pi-test
Let er rip!
C started
C has lock
A started
A waking up B
A grabbing lock
C releasing lock
A has lock
A released lock
A exits
B started
Stopping B
wait for A
wait for B
B exits
wait for C
C no longer has lock
C exits
Priority inheritance worked like a charm!
```

Priority inheritance tracing

```
# trace-cmd show
[..]
#           TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#              | |         |   |||||     |          |
         pi-test-27159   [000] ..... 206947.004556: tracing_mark_write: Let er rip!
         pi-test-27159   [000] d..2. 206947.004579: sched_switch: prev_comm=pi-test prev_pid=27159 prev_prio=93 prev_state=S ==> next_comm=pi
         pi-test-27162   [000] d..2. 206947.004587: sched_waking: comm=pi-test pid=27159 prio=93 target_cpu=000
         pi-test-27162   [000] d..2. 206947.004589: sched_switch: prev_comm=pi-test prev_pid=27162 prev_prio=97 prev_state=R+ ==> next_comm=p
         pi-test-27159   [000] d..2. 206947.004594: sched_switch: prev_comm=pi-test prev_pid=27159 prev_prio=93 prev_state=S ==> next_comm=pi
         pi-test-27162   [000] ..... 206947.004618: tracing_mark_write: C started
         pi-test-27162   [000] ..... 206947.004629: tracing_mark_write: C has lock
         pi-test-27162   [000] d..2. 206947.004637: sched_switch: prev_comm=pi-test prev_pid=27162 prev_prio=97 prev_state=S ==> next_comm=pi
         pi-test-27161   [000] d..2. 206947.004647: sched_switch: prev_comm=pi-test prev_pid=27161 prev_prio=95 prev_state=S ==> next_comm=pi
         pi-test-27160   [000] d..2. 206947.004654: sched_waking: comm=pi-test pid=27162 prio=97 target_cpu=000
         pi-test-27160   [000] ..... 206947.004677: tracing_mark_write: A started
         pi-test-27160   [000] ..... 206947.004686: tracing_mark_write: A waking up B
         pi-test-27160   [000] d..2. 206947.004695: sched_waking: comm=pi-test pid=27161 prio=95 target_cpu=000
         pi-test-27160   [000] ..... 206947.004698: tracing_mark_write: A grabbing lock
         pi-test-27160   [000] d..3. 206947.004708: sched_pi_setprio: comm=pi-test pid=27162 oldprio=97 newprio=94
         pi-test-27160   [000] d..2. 206947.004712: sched_switch: prev_comm=pi-test prev_pid=27160 prev_prio=94 prev_state=S ==> next_comm=pi
         pi-test-27162   [000] d..2. 206947.956144: sched_switch: prev_comm=pi-test prev_pid=27162 prev_prio=94 prev_state=R ==> next_comm=kw
          <idle>-0       [000] d..2. 206948.004549: sched_switch: prev_comm=swapper/0 prev_pid=0 prev_prio=120 prev_state=R ==> next_comm=pi-
         pi-test-27162   [000] ..... 206948.606874: tracing_mark_write: C releasing lock
         pi-test-27162   [000] d..3. 206948.606885: sched_pi_setprio: comm=pi-test pid=27162 oldprio=94 newprio=97
         pi-test-27162   [000] dN.3. 206948.606887: sched_waking: comm=pi-test pid=27160 prio=94 target_cpu=000
         pi-test-27162   [000] d..2. 206948.606889: sched_switch: prev_comm=pi-test prev_pid=27162 prev_prio=97 prev_state=R+ ==> next_comm=p
         pi-test-27160   [000] ..... 206948.606892: tracing_mark_write: A has lock
         pi-test-27160   [000] ..... 206948.606896: tracing_mark_write: A released lock
[..]
```

Instances

- Ideally, you should not have tools using the main tracing directory

Instances

- Ideally, you should not have tools using the main tracing directory
    - Main directory is best for interactive users

Instances

- Ideally, you should not have tools using the main tracing directory
  - Main directory is best for interactive users
- Create an “instance” and use that instead

Instances

- Ideally, you should not have tools using the main tracing directory
    - Main directory is best for interactive users
- Create an “instance” and use that instead
- Instances do not affect the main directory nor other instances

Instances

```
# cd /sys/kernel/tracing

# mkdir instances/foo
```

Instances

```
# cd /sys/kernel/tracing

# mkdir instances/foo
# ls instances/foo
available_tracers     events                  set_event_pid             timestamp_mode     trace_pipe
buffer_percent        free_buffer             set_ftrace_filter         trace              tracing_cpumask
buffer_size_kb        options                 set_ftrace_notrace        trace_clock        tracing_max_latency
buffer_total_size_kb  per_cpu                 set_ftrace_notrace_pid    trace_marker       tracing_on
current_tracer        set_event               set_ftrace_pid            trace_marker_raw
error_log             set_event_notrace_pid   snapshot                  trace_options
```

Instances

```
# cd /sys/kernel/tracing

# mkdir instances/foo
# ls instances/foo
available_tracers     events                   set_event_pid             timestamp_mode     trace_pipe
buffer_percent        free_buffer              set_ftrace_filter         trace              tracing_cpumask
buffer_size_kb        options                  set_ftrace_notrace        trace_clock        tracing_max_latency
buffer_total_size_kb  per_cpu                  set_ftrace_notrace_pid    trace_marker       tracing_on
current_tracer        set_event                set_ftrace_pid            trace_marker_raw
error_log             set_event_notrace_pid    snapshot                  trace_options

# echo 1 > instances/foo/events/sched/sched_switch/enable
```

Instances

```
# cd /sys/kernel/tracing

# mkdir instances/foo
# ls instances/foo
available_tracers     events                 set_event_pid           timestamp_mode     trace_pipe
buffer_percent        free_buffer            set_ftrace_filter       trace              tracing_cpumask
buffer_size_kb        options                set_ftrace_notrace      trace_clock        tracing_max_latency
buffer_total_size_kb  per_cpu                set_ftrace_notrace_pid  trace_marker       tracing_on
current_tracer        set_event              set_ftrace_pid          trace_marker_raw
error_log             set_event_notrace_pid  snapshot                trace_options

# echo 1 > instances/foo/events/sched/sched_switch/enable

# cat trace | tail -2
#           TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#              | |         |   |||||     |         |
```

Instances

```
# cd /sys/kernel/tracing

# mkdir instances/foo
# ls instances/foo
available_tracers     events                 set_event_pid            timestamp_mode     trace_pipe
buffer_percent        free_buffer            set_ftrace_filter        trace              tracing_cpumask
buffer_size_kb        options                set_ftrace_notrace       trace_clock        tracing_max_latency
buffer_total_size_kb  per_cpu                set_ftrace_notrace_pid   trace_marker       tracing_on
current_tracer        set_event              set_ftrace_pid           trace_marker_raw
error_log             set_event_notrace_pid  snapshot                 trace_options

# echo 1 > instances/foo/events/sched/sched_switch/enable

# cat trace | tail -2
#           TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#              | |         |   |||||     |         |

# cat instance/foo/trace | tail -2
          <idle>-0       [005] d..2. 109628.884960: sched_switch: prev_comm=swapper/5 prev_pid=0 prev_prio=120 prev_state=R ==> next
            tail-19861   [005] d..2. 109628.884966: sched_switch: prev_comm=tail prev_pid=19861 prev_prio=120 prev_state=S ==> next_
```

Priority inheritance example using instance (pi-test.c)

```
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <unistd.h>
#define __USE_GNU
#include <sys/syscall.h>
#include <pthread.h>
#include <sched.h>
#include <tracefs.h>

static struct tracefs_instance *instance;

static void trace(const char *fmt, ...)
{
	va_list ap, ap2;

	va_start(ap, fmt);
	va_copy(ap, ap2);
	tracefs_vprintf(instance, fmt, ap);
	vprintf(fmt, ap2);
	va_end(ap2);
	va_end(ap);
}
```

Priority inheritance example using instance (pi-test.c)

```
int main (int argc, char **argv)
{
        pthread_mutexattr_t attr;
        struct tep_handle *tep;
        struct tep_event *sched_waking;
        cpu_set_t cpumask;
        pthread_t A,B,C;
        int secs = SLEEP_SECS;
        char pid[24];

        CPU_ZERO(&cpumask);
        CPU_SET(0, &cpumask);
        sched_setaffinity(0, sizeof(cpumask), &cpumask);

        sprintf(pid, "%ld", gettid());

        instance = tracefs_instance_create("pi-test");

        tracefs_print_init(instance);
        tracefs_event_disable(instance, NULL, NULL);
        tracefs_tracer_clear(instance);
        tracefs_instance_file_write(instance, "trace", "");
        tracefs_instance_file_write(instance, "set_event_pid", pid);
        tracefs_option_enable(instance, TRACEFS_OPTION_EVENT_FORK);
```

Priority inheritance example using instance (pi-test.c)

```
      tep = tracefs_local_events(NULL);
      sched_waking = tep_find_event_by_name(tep, "sched", "sched_waking");
      tracefs_event_filter_apply(instance, sched_waking, "comm == \"pi-test\"");
      tracefs_event_enable(instance, "sched", "sched_waking");
      tracefs_event_enable(instance, "sched", "sched_switch");
      tracefs_event_enable(instance, "sched", "sched_pi_setprio");

[..]
      tracefs_trace_on(instance);
      trace("Let er rip!\n");
      pthread_barrier_wait(&start_C);

      sleep(secs);

      trace("Stopping B\n");
      B_spins_a_long_time = false;

      trace("wait for A\n");
      pthread_join(A, NULL);
      trace("wait for B\n");
      pthread_join(B, NULL);
      trace("wait for C\n");
      pthread_join(C, NULL);

      tracefs_trace_off(instance);
      tracefs_print_close(instance);
```

Priority inheritance tracing with instances

```
# gcc -o pi-test -g -Wall pi-test.c  `pkg-config --cflags --libs libtracefs` -lpthread
# ./pi-test
Let er rip!
C started
C has lock
A started
A waking up B
A grabbing lock
C releasing lock
A has lock
A released lock
A exits
B started
Stopping B
wait for A
wait for B
B exits
wait for C
C no longer has lock
C exits
Priority inheritance worked like a charm!
```

Priority inheritance tracing with instances

```
# trace-cmd show
# tracer: nop
#
# entries-in-buffer/entries-written: 0/0   #P:8
#
#                                _-----=> irqs-off/BH-disabled
#                               / _----=> need-resched
#                              | / _---=> hardirq/softirq
#                              || / _--=> preempt-depth
#                              ||| / _-=> migrate-disable
#                              |||| /     delay
#           TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#              | |         |   |||||     |         |
```

Priority inheritance tracing with instances

```
# trace-cmd show -B pi-test
[..]
#           TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#              | |         |   |||||     |          |
         pi-test-27356   [000] ..... 208307.568339: tracing_mark_write: Let er rip!
         pi-test-27356   [000] d..3. 208307.568351: sched_waking: comm=kworker/u16:2 pid=27015 prio=120 target_cpu=005
         pi-test-27356   [000] d..2. 208307.568361: sched_switch: prev_comm=pi-test prev_pid=27356 prev_prio=93 prev_state=
         pi-test-27359   [000] d..2. 208307.568369: sched_waking: comm=pi-test pid=27356 prio=93 target_cpu=000
         pi-test-27359   [000] d..2. 208307.568371: sched_switch: prev_comm=pi-test prev_pid=27359 prev_prio=97 prev_state=
         pi-test-27356   [000] d..2. 208307.568376: sched_switch: prev_comm=pi-test prev_pid=27356 prev_prio=93 prev_state=
         pi-test-27359   [000] ..... 208307.568407: tracing_mark_write: C started
         pi-test-27359   [000] d..3. 208307.568410: sched_waking: comm=kworker/u16:2 pid=27015 prio=120 target_cpu=005
         pi-test-27359   [000] ..... 208307.568415: tracing_mark_write: C has lock
         pi-test-27359   [000] d..3. 208307.568417: sched_waking: comm=kworker/u16:2 pid=27015 prio=120 target_cpu=005
         pi-test-27359   [000] d..2. 208307.568421: sched_switch: prev_comm=pi-test prev_pid=27359 prev_prio=97 prev_state=
         pi-test-27358   [000] d..2. 208307.568428: sched_switch: prev_comm=pi-test prev_pid=27358 prev_prio=95 prev_state=
         pi-test-27357   [000] d..2. 208307.568433: sched_waking: comm=pi-test pid=27359 prio=97 target_cpu=000
         pi-test-27357   [000] ..... 208307.568449: tracing_mark_write: A started
         pi-test-27357   [000] d..3. 208307.568451: sched_waking: comm=kworker/u16:2 pid=27015 prio=120 target_cpu=005
         pi-test-27357   [000] ..... 208307.568454: tracing_mark_write: A waking up B
         pi-test-27357   [000] d..3. 208307.568456: sched_waking: comm=kworker/u16:2 pid=27015 prio=120 target_cpu=005
         pi-test-27357   [000] d..2. 208307.568458: sched_waking: comm=pi-test pid=27358 prio=95 target_cpu=000
         pi-test-27357   [000] ..... 208307.568459: tracing_mark_write: A grabbing lock
         pi-test-27357   [000] d..3. 208307.568461: sched_waking: comm=kworker/u16:2 pid=27015 prio=120 target_cpu=005
         pi-test-27357   [000] d..3. 208307.568466: sched_pi_setprio: comm=pi-test pid=27359 oldprio=97 newprio=94
[..]
```

trace-cmd is your friend

- Installed on most distributions

trace-cmd is your friend

- Installed on most distributions
- Handles the mounting of tracefs

trace-cmd is your friend

- Installed on most distributions
- Handles the mounting of tracefs
- Can start and read tracing

trace-cmd is your friend

- Installed on most distributions
- Handles the mounting of tracefs
- Can start and read tracing
- Can record tracing to a file
  - Default: “trace.dat”, but can be a customized name

trace-cmd is your friend

- Installed on most distributions
- Handles the mounting of tracefs
- Can start and read tracing
- Can record tracing to a file
  - Default: “trace.dat”, but can be a customized name
- Can extract existing data into trace.dat

trace-cmd is your friend

- Installed on most distributions
- Handles the mounting of tracefs
- Can start and read tracing
- Can record tracing to a file
  - Default: “trace.dat”, but can be a customized name
- Can extract existing data into trace.dat
- libtracecmd can be used to read the trace.dat file from other applications

Extracting the data

```
# trace-cmd extract -B pi-test
```

Extracting the data

```
# trace-cmd extract -B pi-test
# trace-cmd report
version = 7
cpus=8
pi-test:          pi-test-31150 [000] 256854.468365: print:                tracing_mark_write: Let er rip!
pi-test:          pi-test-31150 [000] 256854.468377: sched_waking:         comm=kworker/u16:0 pid=31080 prio=120 target_cpu=004
pi-test:          pi-test-31150 [000] 256854.468388: sched_switch:         pi-test:31150 [93] S ==> pi-test:31153 [120]
pi-test:          pi-test-31153 [000] 256854.468398: sched_waking:         comm=pi-test pid=31150 prio=93 target_cpu=000
pi-test:          pi-test-31153 [000] 256854.468401: sched_switch:         pi-test:31153 [97] R ==> pi-test:31150 [93]
pi-test:          pi-test-31150 [000] 256854.468406: sched_switch:         pi-test:31150 [93] S ==> pi-test:31153 [97]
pi-test:          pi-test-31153 [000] 256854.468428: print:                tracing_mark_write: C started
pi-test:          pi-test-31153 [000] 256854.468431: sched_waking:         comm=kworker/u16:0 pid=31080 prio=120 target_cpu=004
pi-test:          pi-test-31153 [000] 256854.468438: print:                tracing_mark_write: C has lock
pi-test:          pi-test-31153 [000] 256854.468440: sched_waking:         comm=kworker/u16:0 pid=31080 prio=120 target_cpu=004
pi-test:          pi-test-31153 [000] 256854.468446: sched_switch:         pi-test:31153 [97] S ==> pi-test:31152 [120]
pi-test:          pi-test-31152 [000] 256854.468456: sched_switch:         pi-test:31152 [95] S ==> pi-test:31151 [120]
pi-test:          pi-test-31151 [000] 256854.468461: sched_waking:         comm=pi-test pid=31153 prio=97 target_cpu=000
pi-test:          pi-test-31151 [000] 256854.468477: print:                tracing_mark_write: A started
pi-test:          pi-test-31151 [000] 256854.468480: sched_waking:         comm=kworker/u16:0 pid=31080 prio=120 target_cpu=004
pi-test:          pi-test-31151 [000] 256854.468484: print:                tracing_mark_write: A waking up B
pi-test:          pi-test-31151 [000] 256854.468488: sched_waking:         comm=kworker/u16:0 pid=31080 prio=120 target_cpu=004
pi-test:          pi-test-31151 [000] 256854.468491: sched_waking:         comm=pi-test pid=31152 prio=95 target_cpu=000
pi-test:          pi-test-31151 [000] 256854.468493: print:                tracing_mark_write: A grabbing lock
pi-test:          pi-test-31151 [000] 256854.468496: sched_waking:         comm=kworker/u16:0 pid=31080 prio=120 target_cpu=004
pi-test:          pi-test-31151 [000] 256854.468503: sched_pi_setprio:     comm=pi-test pid=31153 oldprio=97 newprio=94
pi-test:          pi-test-31151 [000] 256854.468506: sched_switch:         pi-test:31151 [94] S ==> pi-test:31153 [94]
pi-test:          pi-test-31153 [000] 256854.480160: sched_waking:         comm=kworker/0:1 pid=28353 prio=120 target_cpu=000
pi-test:          pi-test-31153 [000] 256854.482159: sched_waking:         comm=kcompactd0 pid=82 prio=120 target_cpu=003
pi-test:          pi-test-31153 [000] 256854.604157: sched_waking:         comm=qemu-system-x86 pid=29967 prio=120 target_cpu=003
pi-test:          pi-test-31153 [000] 256854.802157: sched_waking:         comm=kworker/u16:0 pid=31080 prio=120 target_cpu=004
pi-test:          pi-test-31153 [000] 256854.840266: sched_waking:         comm=kworker/0:1H pid=9 prio=100 target_cpu=000
[..]
```

Reading a trace.dat file (read-trace.c)

```
#define _GNU_SOURCE
#include <stdlib.h>
#include <unistd.h>
#include <sched.h>
#include <errno.h>
#include <trace-cmd.h>

static int print_record(struct tep_handle *tep, struct tep_record *record, struct trace_seq *seq);
static struct tep_record *get_next_record(struct tep_handle *tep, struct tracecmd_input *handle);
```

Reading a trace.dat file (read-trace.c)

```
int main(int argc, char **argv) {
        struct tracecmd_input *handle, *pi_handle;
        struct tep_record *record;
        struct tep_handle *tep;
        struct trace_seq seq;
        char *file = "trace.dat";
        int i, nr_instances;

        if (argc > 1)
                file = argv[1];

        tracecmd_set_loglevel(TEP_LOG_CRITICAL);

        handle = tracecmd_open(file, 0);
        nr_instances = tracecmd_buffer_instances(handle);
        for (i = 0; i < nr_instances; i++) {
                if (strcmp("pi-test", tracecmd_buffer_instance_name(handle, i)) == 0)
                        break;
        }
        pi_handle = tracecmd_buffer_instance_handle(handle, i);

        tep = tracecmd_get_tep(pi_handle);
        trace_seq_init(&seq);

        while ((record = get_next_record(tep, pi_handle)) != NULL) {
                print_record(tep, record, &seq);
                tracecmd_free_record(record);
        }
        trace_seq_destroy(&seq);
        return 0;
}
```

Reading a trace.dat file (read-trace.c)

```
int main(int argc, char **argv) {
        struct tracecmd_input *handle, *pi_handle;
        struct tep_record *record;
        struct tep_handle *tep;
        struct trace_seq seq;
        char *file = "trace.dat";
        int i, nr_instances;

        if (argc > 1)
                file = argv[1];

        tracecmd_set_loglevel(TEP_LOG_CRITICAL);

        handle = tracecmd_open(file, 0);
        nr_instances = tracecmd_buffer_instances(handle);
        for (i = 0; i < nr_instances; i++) {
                if (strcmp("pi-test", tracecmd_buffer_instance_name(handle, i)) == 0)
                        break;
        }
        pi_handle = tracecmd_buffer_instance_handle(handle, i);

        tep = tracecmd_get_tep(pi_handle);
        trace_seq_init(&seq);

        while ((record = get_next_record(tep, pi_handle)) != NULL) {
                print_record(tep, record, &seq);
                tracecmd_free_record(record);
        }
        trace_seq_destroy(&seq);
        return 0;
}
```

Reading a trace.dat file (read-trace.c)

```
int main(int argc, char **argv) {
        struct tracecmd_input *handle, *pi_handle;
        struct tep_record *record;
        struct tep_handle *tep;
        struct trace_seq seq;
        char *file = "trace.dat";
        int i, nr_instances;

        if (argc > 1)
                file = argv[1];

        tracecmd_set_loglevel(TEP_LOG_CRITICAL);

        handle = tracecmd_open(file, 0);
        nr_instances = tracecmd_buffer_instances(handle);
        for (i = 0; i < nr_instances; i++) {
                if (strcmp("pi-test", tracecmd_buffer_instance_name(handle, i)) == 0)
                        break;
        }
        pi_handle = tracecmd_buffer_instance_handle(handle, i);

        tep = tracecmd_get_tep(pi_handle);
        trace_seq_init(&seq);

        while ((record = get_next_record(tep, pi_handle)) != NULL) {
                print_record(tep, record, &seq);
                tracecmd_free_record(record);
        }
        trace_seq_destroy(&seq);
        return 0;
}
```

Reading a trace.dat file (read-trace.c)

```
int main(int argc, char **argv) {
        struct tracecmd_input *handle, *pi_handle;
        struct tep_record *record;
        struct tep_handle *tep;
        struct trace_seq seq;
        char *file = "trace.dat";
        int i, nr_instances;

        if (argc > 1)
                file = argv[1];

        tracecmd_set_loglevel(TEP_LOG_CRITICAL);

        handle = tracecmd_open(file, 0);
        nr_instances = tracecmd_buffer_instances(handle);
        for (i = 0; i < nr_instances; i++) {
                if (strcmp("pi-test", tracecmd_buffer_instance_name(handle, i)) == 0)
                        break;
        }
        pi_handle = tracecmd_buffer_instance_handle(handle, i);

        tep = tracecmd_get_tep(pi_handle);
        trace_seq_init(&seq);

        while ((record = get_next_record(tep, pi_handle)) != NULL) {
                print_record(tep, record, &seq);
                tracecmd_free_record(record);
        }
        trace_seq_destroy(&seq);
        return 0;
}
```

Reading a trace.dat file (read-trace.c)

```
int main(int argc, char **argv) {
        struct tracecmd_input *handle, *pi_handle;
        struct tep_record *record;
        struct tep_handle *tep;
        struct trace_seq seq;
        char *file = "trace.dat";
        int i, nr_instances;

        if (argc > 1)
                file = argv[1];

        tracecmd_set_loglevel(TEP_LOG_CRITICAL);

        handle = tracecmd_open(file, 0);
        nr_instances = tracecmd_buffer_instances(handle);
        for (i = 0; i < nr_instances; i++) {
                if (strcmp("pi-test", tracecmd_buffer_instance_name(handle, i)) == 0)
                        break;
        }
        pi_handle = tracecmd_buffer_instance_handle(handle, i);

        tep = tracecmd_get_tep(pi_handle);
        trace_seq_init(&seq);

        while ((record = get_next_record(tep, pi_handle)) != NULL) {
                print_record(tep, record, &seq);
                tracecmd_free_record(record);
        }
        trace_seq_destroy(&seq);
        return 0;
}
```

Reading a trace.dat file (read-trace.c)

```c
int main(int argc, char **argv) {
        struct tracecmd_input *handle, *pi_handle;
        struct tep_record *record;
        struct tep_handle *tep;
        struct trace_seq seq;
        char *file = "trace.dat";
        int i, nr_instances;

        if (argc > 1)
                file = argv[1];

        tracecmd_set_loglevel(TEP_LOG_CRITICAL);

        handle = tracecmd_open(file, 0);
        nr_instances = tracecmd_buffer_instances(handle);
        for (i = 0; i < nr_instances; i++) {
                if (strcmp("pi-test", tracecmd_buffer_instance_name(handle, i)) == 0)
                        break;
        }
        pi_handle = tracecmd_buffer_instance_handle(handle, i);

        tep = tracecmd_get_tep(pi_handle);
        trace_seq_init(&seq);

        while ((record = get_next_record(tep, pi_handle)) != NULL) {
                print_record(tep, record, &seq);
                tracecmd_free_record(record);
        }
        trace_seq_destroy(&seq);
        return 0;
}
```

Reading a trace.dat file (read-trace.c)

```c
static struct tep_record *get_next_record(struct tep_handle *tep,
                                          struct tracecmd_input *handle)
{
        static struct tep_record **cpu_records;
        struct tep_record *record;
        unsigned long long ts = -1ULL;
        int nr_cpus;
        int next = -1;
        int i;

        nr_cpus = tep_get_cpus(tep);

        if (!cpu_records) {
                cpu_records = calloc(nr_cpus, sizeof(*cpu_records));
                for (i = 0; i < nr_cpus; i++)
                        cpu_records[i] = tracecmd_read_cpu_first(handle, i);
        }

        for (i = 0; i < nr_cpus; i++) {
                if (cpu_records[i] && cpu_records[i]->ts < ts)
                        next = i;
        }
        if (next < 0)
                return NULL;

        record = cpu_records[next];
        cpu_records[next] = tracecmd_read_data(handle, next);

        return record;
}
```

Reading a trace.dat file (read-trace.c)

```
static struct tep_record *get_next_record(struct tep_handle *tep,
                                          struct tracecmd_input *handle)
{
        static struct tep_record **cpu_records;
        struct tep_record *record;
        unsigned long long ts = -1ULL;
        int nr_cpus;
        int next = -1;
        int i;

        nr_cpus = tep_get_cpus(tep);

        if (!cpu_records) {
                cpu_records = calloc(nr_cpus, sizeof(*cpu_records));
                for (i = 0; i < nr_cpus; i++)
                        cpu_records[i] = tracecmd_read_cpu_first(handle, i);
        }

        for (i = 0; i < nr_cpus; i++) {
                if (cpu_records[i] && cpu_records[i]->ts < ts)
                        next = i;
        }
        if (next < 0)
                return NULL;

        record = cpu_records[next];
        cpu_records[next] = tracecmd_read_data(handle, next);

        return record;
}
```

Reading a trace.dat file (read-trace.c)

```
static struct tep_record *get_next_record(struct tep_handle *tep,
                                          struct tracecmd_input *handle)
{
        static struct tep_record **cpu_records;
        struct tep_record *record;
        unsigned long long ts = -1ULL;
        int nr_cpus;
        int next = -1;
        int i;

        nr_cpus = tep_get_cpus(tep);

        if (!cpu_records) {
                cpu_records = calloc(nr_cpus, sizeof(*cpu_records));
                for (i = 0; i < nr_cpus; i++)
                        cpu_records[i] = tracecmd_read_cpu_first(handle, i);
        }

        for (i = 0; i < nr_cpus; i++) {
                if (cpu_records[i] && cpu_records[i]->ts < ts)
                        next = i;
        }
        if (next < 0)
                return NULL;

        record = cpu_records[next];
        cpu_records[next] = tracecmd_read_data(handle, next);

        return record;
}
```

Reading a trace.dat file (read-trace.c)

```c
static struct tep_record *get_next_record(struct tep_handle *tep,
                                          struct tracecmd_input *handle)
{
        static struct tep_record **cpu_records;
        struct tep_record *record;
        unsigned long long ts = -1ULL;
        int nr_cpus;
        int next = -1;
        int i;

        nr_cpus = tep_get_cpus(tep);

        if (!cpu_records) {
                cpu_records = calloc(nr_cpus, sizeof(*cpu_records));
                for (i = 0; i < nr_cpus; i++)
                        cpu_records[i] = tracecmd_read_cpu_first(handle, i);
        }

        for (i = 0; i < nr_cpus; i++) {
                if (cpu_records[i] && cpu_records[i]->ts < ts)
                        next = i;
        }
        if (next < 0)
                return NULL;

        record = cpu_records[next];
        cpu_records[next] = tracecmd_read_data(handle, next);

        return record;
}
```

Reading a trace.dat file (read-trace.c)

```
static struct tep_record *get_next_record(struct tep_handle *tep,
                                          struct tracecmd_input *handle)
{
        static struct tep_record **cpu_records;
        struct tep_record *record;
        unsigned long long ts = -1ULL;
        int nr_cpus;
        int next = -1;
        int i;

        nr_cpus = tep_get_cpus(tep);

        if (!cpu_records) {
                cpu_records = calloc(nr_cpus, sizeof(*cpu_records));
                for (i = 0; i < nr_cpus; i++)
                        cpu_records[i] = tracecmd_read_cpu_first(handle, i);
        }

        for (i = 0; i < nr_cpus; i++) {
                if (cpu_records[i] && cpu_records[i]->ts < ts)
                        next = i;
        }
        if (next < 0)
                return NULL;

        record = cpu_records[next];
        cpu_records[next] = tracecmd_read_data(handle, next);

        return record;
}
```

Reading a trace.dat file (read-trace.c)

```
static int print_record(struct tep_handle *tep, struct tep_record *record,
                        struct trace_seq *seq)
{
        int cpu = record->cpu;

        trace_seq_reset(seq);
        tep_print_event(tep, seq, record, "%6.1000d", TEP_PRINT_TIME);
        trace_seq_printf(seq, " [%03d] ", cpu);
        tep_print_event(tep, seq, record, "%s-%d %s %s\n",
                        TEP_PRINT_COMM, TEP_PRINT_PID,
                        TEP_PRINT_NAME, TEP_PRINT_INFO);
        trace_seq_do_printf(seq);
        return 0;
}
```

Reading the trace file

```
# gcc -o read-trace -g -Wall read-trace.c  `pkg-config --cflags --libs libtracecmd`
```

Reading the trace file

```
# gcc -o read-trace -g -Wall read-trace.c  `pkg-config --cflags --libs libtracecmd`
# ./read-trace
274187.521073 [000] pi-test-33358 print tracing_mark_write: Let er rip!

274187.521098 [000] pi-test-33358 sched_switch pi-test:33358 [93] S ==> pi-test:33361 [120]
274187.521107 [000] pi-test-33361 sched_waking comm=pi-test pid=33358 prio=93 target_cpu=000
274187.521109 [000] pi-test-33361 sched_switch pi-test:33361 [97] R ==> pi-test:33358 [93]
274187.521115 [000] pi-test-33358 sched_switch pi-test:33358 [93] S ==> pi-test:33361 [97]
274187.521135 [000] pi-test-33361 print tracing_mark_write: C started

274187.521143 [000] pi-test-33361 print tracing_mark_write: C has lock

274187.521150 [000] pi-test-33361 sched_switch pi-test:33361 [97] S ==> pi-test:33360 [120]
274187.521157 [000] pi-test-33360 sched_switch pi-test:33360 [95] S ==> pi-test:33359 [120]
274187.521161 [000] pi-test-33359 sched_waking comm=pi-test pid=33361 prio=97 target_cpu=000
274187.521177 [000] pi-test-33359 print tracing_mark_write: A started

274187.521183 [000] pi-test-33359 print tracing_mark_write: A waking up B

274187.521186 [000] pi-test-33359 sched_waking comm=pi-test pid=33360 prio=95 target_cpu=000
274187.521188 [000] pi-test-33359 print tracing_mark_write: A grabbing lock

274187.521195 [000] pi-test-33359 sched_pi_setprio comm=pi-test pid=33361 oldprio=97 newprio=94
274187.521197 [000] pi-test-33359 sched_switch pi-test:33359 [94] S ==> pi-test:33361 [94]
274188.385044 [000] pi-test-33361 sched_switch pi-test:33361 [94] R ==> migration/0:19 [0]
274188.385047 [000] migration/0-19 sched_switch migration/0:19 [0] S ==> pi-test:33361 [94]
274188.473039 [000] pi-test-33361 sched_switch pi-test:33361 [94] R ==> kworker/0:2:31996 [120]
274188.521065 [000] <idle>-0 sched_switch swapper/0:0 [120] R ==> pi-test:33361 [94]
274189.120523 [000] pi-test-33361 print tracing_mark_write: C releasing lock
[..]
```

Writing structures into the ring buffer from the application

- **tracefs_binary_init**()
  - Pre-initalize the tracefs printing to not need to do that during hot paths of the application

Writing structures into the ring buffer from the application

- **tracefs_binary_init**()
  - Pre-initalize the tracefs printing to not need to do that during hot paths of the application
- **tracefs_binary_write**()
  - Writes into the ring buffer (will call tracefs_binary_init() if it is not already initialized)

Writing structures into the ring buffer from the application

- **tracefs_binary_init**()
  - Pre-initalize the tracefs printing to not need to do that during hot paths of the application
- **tracefs_binary_write**()
  - Writes into the ring buffer (will call tracefs_binary_init() if it is not already initialized)
- **tracefs_binary_close**()
  - Cleans up the open file descriptors from tracefs_binary_init()

Writing structures into the ring buffer (pi-test.c)

```
#define __USE_GNU
#include <sys/syscall.h>
#include <pthread.h>
#include <sched.h>
#include <tracefs.h>

enum tasks {
        TASK_A,
        TASK_B,
        TASK_C,
};

enum pi_states {
        PI_START,
        PI_GRABBING_LOCK,
        PI_HAS_LOCK,
        PI_RELEASING_LOCK,
        PI_RELEASED_LOCK,
        PI_FINISH,
};

struct pi_state {
        unsigned int    id;
        unsigned int    tid;
        enum tasks      task;
        enum pi_states  state;
};

#define PI_ID 1991

static void trace(const char *fmt, ...)
{
        va_list ap, ap2;
```

Writing structures into the ring buffer (pi-test.c)

```
static void *thread_A(void *arg)
{
        struct pi_state state = { .id = PI_ID, .task = TASK_A };

        state.tid = gettid();
        set_prio(A_PRIO);

        pthread_barrier_wait(&start_A);
        state.state = PI_START;
        tracefs_binary_write(instance, &state, sizeof(state));
        pthread_barrier_wait(&start_B);

        state.state = PI_GRABBING_LOCK;
        tracefs_binary_write(instance, &state, sizeof(state));

        pthread_mutex_lock(&L_lock);
        if (B_ran_a_lot)
                PI_has_failed = true;

        state.state = PI_HAS_LOCK;
        tracefs_binary_write(instance, &state, sizeof(state));

        state.state = PI_RELEASING_LOCK;
        tracefs_binary_write(instance, &state, sizeof(state));

        pthread_mutex_unlock(&L_lock);

        state.state = PI_RELEASED_LOCK;
        tracefs_binary_write(instance, &state, sizeof(state));
        state.state = PI_FINISH;
        tracefs_binary_write(instance, &state, sizeof(state));

        return NULL;
}
```

Writing structures into the ring buffer (pi-test.c)

```
static void *thread_B(void *arg)
{
	struct pi_state state = { .id = PI_ID, .task = TASK_B };

	state.tid = gettid();

	set_prio(B_PRIO);

	pthread_barrier_wait(&start_B);

	state.state = PI_START;
	tracefs_binary_write(instance, &state, sizeof(state));

	while (B_spins_a_long_time)
		barrier();

	B_ran_a_lot = 1;

	state.state = PI_FINISH;
	tracefs_binary_write(instance, &state, sizeof(state));

	return NULL;
}
```

Writing structures into the ring buffer (pi-test.c)

```c
static void *thread_C(void *arg)
{
        struct pi_state state = { .id = PI_ID, .task = TASK_C };
        unsigned long long i;

        state.tid = gettid();
        set_prio(C_PRIO);

        pthread_barrier_wait(&start_C);
        state.state = PI_START;
        tracefs_binary_write(instance, &state, sizeof(state));

        state.state = PI_GRABBING_LOCK;
        tracefs_binary_write(instance, &state, sizeof(state));
        pthread_mutex_lock(&L_lock);

        state.state = PI_HAS_LOCK;
        tracefs_binary_write(instance, &state, sizeof(state));
        pthread_barrier_wait(&start_A);

        /* spin a little */
        for (i=0; i < 1000000000; i++)
                barrier();

        state.state = PI_RELEASING_LOCK;
        tracefs_binary_write(instance, &state, sizeof(state));
        pthread_mutex_unlock(&L_lock);

        state.state = PI_RELEASED_LOCK;
        tracefs_binary_write(instance, &state, sizeof(state));
        state.state = PI_FINISH;
        tracefs_binary_write(instance, &state, sizeof(state));

        return NULL;
}
```

Writing structures into the ring buffer (pi-test.c)

```
int main (int argc, char **argv)
{
        pthread_mutexattr_t attr;
        struct tep_handle *tep;
        struct tep_event *sched_waking;
        cpu_set_t cpumask;
        pthread_t A,B,C;
        int secs = SLEEP_SECS;
        char pid[24];

        CPU_ZERO(&cpumask);
        CPU_SET(0, &cpumask);
        sched_setaffinity(0, sizeof(cpumask), &cpumask);

        sprintf(pid, "%ld", gettid());

        instance = tracefs_instance_create("pi-test");

        tracefs_binary_init(instance);
        tracefs_print_init(instance);
        tracefs_event_disable(instance, NULL, NULL);
[..]
        tracefs_trace_off(instance);
        tracefs_print_close(instance);
        tracefs_binary_close(instance);

        if (PI_has_failed)
                printf("Priority inheritance failed\n");
        else
                printf("Priority inheritance worked like a charm!\n");

        exit(0);
}
```

Reading structures from the ring buffer (read-trace.c)

```
#define _GNU_SOURCE
#include <stdlib.h>
#include <unistd.h>
#include <sched.h>
#include <errno.h>
#include <trace-cmd.h>

enum tasks {
        TASK_A,
        TASK_B,
        TASK_C,
};

enum pi_states {
        PI_START,
        PI_GRABBING_LOCK,
        PI_HAS_LOCK,
        PI_RELEASING_LOCK,
        PI_RELEASED_LOCK,
        PI_FINISH,
};

struct pi_state {
        unsigned int    id;
        unsigned int    tid;
        enum tasks      task;
        enum pi_states  state;
};

#define PI_ID 1991
```

Reading structures from the ring buffer (read-trace.c)

```
int main(int argc, char **argv) {
        struct tracecmd_input *handle, *pi_handle;
        struct tep_record *record;
        struct tep_event *raw_event;
        struct tep_format_field *id;
        struct tep_handle *tep;
        struct trace_seq seq;
        char *file = "trace.dat";
        int i, nr_instances;

        if (argc > 1)
                file = argv[1];

        tracecmd_set_loglevel(TEP_LOG_CRITICAL);

        handle = tracecmd_open(file, 0);
        nr_instances = tracecmd_buffer_instances(handle);
        for (i = 0; i < nr_instances; i++) {
                if (strcmp("pi-test", tracecmd_buffer_instance_name(handle, i)) == 0)
                        break;
        }
        pi_handle = tracecmd_buffer_instance_handle(handle, i);

        tep = tracecmd_get_tep(pi_handle);
        trace_seq_init(&seq);

        raw_event = tep_find_event_by_name(tep, "ftrace", "raw_data");
        raw_id = raw_event->id;
        id = tep_find_field(raw_event, "id");
        data_offset = id->offset;

        while ((record = get_next_record(tep, pi_handle)) != NULL) {
                print_record(tep, record, &seq);
[..]
```

Reading structures from the ring buffer (read-trace.c)

```
static void process_state(struct pi_state *pstate);

static unsigned int raw_id;
static unsigned int data_offset;

static int print_record(struct tep_handle *tep, struct tep_record *record,
                        struct trace_seq *seq)
{
        struct pi_state *state;
        int cpu = record->cpu;

        if (tep_data_type(tep, record) == raw_id) {
                if (record->size - data_offset >= sizeof(*state)) {
                        state = record->data + data_offset;
                        if (state->id == PI_ID)
                                process_state(state);
                }
                return 0;
        }
        trace_seq_reset(seq);
        tep_print_event(tep, seq, record, "%6.1000d", TEP_PRINT_TIME);
        trace_seq_printf(seq, " [%03d] ", cpu);
        tep_print_event(tep, seq, record, "%s-%d %s %s\n",
                        TEP_PRINT_COMM, TEP_PRINT_PID,
                        TEP_PRINT_NAME, TEP_PRINT_INFO);
        trace_seq_do_printf(seq);
        return 0;
}
```

Reading structures from the ring buffer (read-trace.c)

```
raw_id = raw_event->id;
```

```
static void process_state(struct pi_state *pstate);

static unsigned int raw_id;
static unsigned int data_offset;

static int print_record(struct tep_handle *tep, struct tep_record *record,
                        struct trace_seq *seq)
{
        struct pi_state *s
        int cpu = record->cpu;

        if (tep_data_type(tep, record) == raw_id) {
                if (record->size - data_offset >= sizeof(*state)) {
                        state = record->data + data_offset;
                        if (state->id == PI_ID)
                                process_state(state);
                }
                return 0;
        }
        trace_seq_reset(seq);
        tep_print_event(tep, seq, record, "%6.1000d", TEP_PRINT_TIME);
        trace_seq_printf(seq, " [%03d] ", cpu);
        tep_print_event(tep, seq, record, "%s-%d %s %s\n",
                        TEP_PRINT_COMM, TEP_PRINT_PID,
                        TEP_PRINT_NAME, TEP_PRINT_INFO);
        trace_seq_do_printf(seq);
        return 0;
}
```

Reading structures from the ring buffer (read-trace.c)

```c
static void process_state(struct pi_state *pstate);

static unsigned int raw_id;
static unsigned int data_offset;

static int print_record(struct tep_handle *tep, struct tep_record *record,
                        struct trace_seq *seq)
{
        struct pi_sta
        int cpu = rec

        if (tep_data_type(tep, record) == raw_id) {
                if (record->size - data_offset >= sizeof(*state)) {
                        state = record->data + data_offset;
                        if (state->id == PI_ID)
                                process_state(state);
                }
                return 0;
        }
        trace_seq_reset(seq);
        tep_print_event(tep, seq, record, "%6.1000d", TEP_PRINT_TIME);
        trace_seq_printf(seq, " [%03d] ", cpu);
        tep_print_event(tep, seq, record, "%s-%d %s %s\n",
                        TEP_PRINT_COMM, TEP_PRINT_PID,
                        TEP_PRINT_NAME, TEP_PRINT_INFO);
        trace_seq_do_printf(seq);
        return 0;
}
```

```c
        id = tep_find_field(raw_event, "id");
        data_offset = id->offset;
```

Reading structures from the ring buffer (read-trace.c)

```
static void process_state(struct pi_state *pstate);

static unsigned int raw_id;
static unsigned int data_offset;

static int print_record(struct tep_handle *tep, struct tep_record *record,
                      struct trace_seq *seq)
{
       struct pi_state *state;
       int cpu = record->cpu;

       if (tep_data_type(tep, record) == raw_id) {
               if (record->size - data_offset >= sizeof(*state)) {
                       state = record->data + data_offset;
                       if (state->id == PI_ID)
                               process_state(state);
               }
               return 0;
       }
       trace_seq_reset(seq);
       tep_print_event(tep, seq, record, "%6.1000d", TEP_PRINT_TIME);
       trace_seq_printf(seq, " [%03d] ", cpu);
       tep_print_event(tep, seq, record, "%s-%d %s %s\n",
                       TEP_PRINT_COMM, TEP_PRINT_PID,
                       TEP_PRINT_NAME, TEP_PRINT_INFO);
       trace_seq_do_printf(seq);
       return 0;
}
```

Reading structures from the ring buffer (read-trace.c)

```
static void process_state(struct pi_state *pstate)
{
        const char *task;
        const char *state;

        switch (pstate->task) {
        case TASK_A: task = "Task A"; break;
        case TASK_B: task = "Task B"; break;
        case TASK_C: task = "Task C"; break;
        }

        switch (pstate->state) {
        case PI_START: state = "staring"; break;
        case PI_GRABBING_LOCK: state = "grabbing lock"; break;
        case PI_HAS_LOCK: state = "has lock"; break;
        case PI_RELEASING_LOCK: state = "releasing lock"; break;
        case PI_RELEASED_LOCK: state = "released lock"; break;
        case PI_FINISH: state = "finished"; break;
        }

        printf("[%d] %s %s\n", pstate->tid, task, state);
}
```

Passing binary data via the trace buffer

```
# gcc -o pi-test -g -Wall pi-test.c  `pkg-config --cflags --libs libtracefs`
# ./pi-test
Let er rip!
Stopping B
wait for A
wait for B
wait for C
Priority inheritance worked like a charm!
```

Passing binary data via the trace buffer

```
# gcc -o pi-test -g -Wall pi-test.c  `pkg-config --cflags --libs libtracefs`
# ./pi-test
Let er rip!
Stopping B
wait for A
wait for B
wait for C
Priority inheritance worked like a charm!

# trace-cmd extract -B pi-test
```

Passing binary data via the trace buffer

```
# gcc -o pi-test -g -Wall pi-test.c  `pkg-config --cflags --libs libtracefs`
# ./pi-test
Let er rip!
Stopping B
wait for A
wait for B
wait for C
Priority inheritance worked like a charm!

# trace-cmd extract -B pi-test
# trace-cmd report
version = 7
cpus=8
pi-test:          pi-test-33861 [000] 279566.407884: print:                tracing_mark_write: Let er rip!
pi-test:          pi-test-33861 [000] 279566.407905: sched_switch:         pi-test:33861 [93] S ==> pi-test:33864 [120]
pi-test:          pi-test-33864 [000] 279566.407914: sched_waking:         comm=pi-test pid=33861 prio=93 target_cpu=000
pi-test:          pi-test-33864 [000] 279566.407915: sched_switch:         pi-test:33864 [97] R ==> pi-test:33861 [93]
pi-test:          pi-test-33861 [000] 279566.407920: sched_switch:         pi-test:33861 [93] S ==> pi-test:33864 [97]
pi-test:          pi-test-33864 [000] 279566.407922: raw_data:             # 7c7 48 84 00 00 02 00 00 00 00 00 00 00 64 0a 00 00
pi-test:          pi-test-33864 [000] 279566.407923: raw_data:             # 7c7 48 84 00 00 02 00 00 00 01 00 00 00 63 6b 0a 00
pi-test:          pi-test-33864 [000] 279566.407924: raw_data:             # 7c7 48 84 00 00 02 00 00 00 02 00 00 00 00 00 00 00
pi-test:          pi-test-33864 [000] 279566.407926: sched_switch:         pi-test:33864 [97] S ==> pi-test:33863 [120]
pi-test:          pi-test-33863 [000] 279566.407932: sched_switch:         pi-test:33863 [95] S ==> pi-test:33862 [120]
pi-test:          pi-test-33862 [000] 279566.407937: sched_waking:         comm=pi-test pid=33864 prio=97 target_cpu=000
pi-test:          pi-test-33862 [000] 279566.407939: raw_data:             # 7c7 46 84 00 00 00 00 00 00 00 00 00 00 6b 69 6e 67
pi-test:          pi-test-33862 [000] 279566.407940: sched_waking:         comm=pi-test pid=33863 prio=95 target_cpu=000
pi-test:          pi-test-33862 [000] 279566.407941: raw_data:             # 7c7 46 84 00 00 00 00 00 00 01 00 00 00 41 20 67 72
pi-test:          pi-test-33862 [000] 279566.407944: sched_pi_setprio:     comm=pi-test pid=33864 oldprio=97 newprio=94
pi-test:          pi-test-33862 [000] 279566.407947: sched_switch:         pi-test:33862 [94] S ==> pi-test:33864 [94]
pi-test:          pi-test-33864 [000] 279567.359532: sched_switch:         pi-test:33864 [94] R ==> kworker/0:2:31996 [120]
pi-test:           <idle>-0     [000] 279567.407868: sched_switch:         swapper/0:0 [120] R ==> pi-test:33864 [94]
[,,]
```

Passing binary data via the trace buffer

```
# gcc -o read-trace -g -Wall read-trace.c  `pkg-config --cflags --libs libtracecmd
```

Passing binary data via the trace buffer

```
# gcc -o read-trace -g -Wall read-trace.c  `pkg-config --cflags --libs libtracecmd`
# ./read-trace
279566.407884 [000] pi-test-33861 print tracing_mark_write: Let er rip!

279566.407905 [000] pi-test-33861 sched_switch pi-test:33861 [93] S ==> pi-test:33864 [120]
279566.407914 [000] pi-test-33864 sched_waking comm=pi-test pid=33861 prio=93 target_cpu=000
279566.407915 [000] pi-test-33864 sched_switch pi-test:33864 [97] R ==> pi-test:33861 [93]
279566.407920 [000] pi-test-33861 sched_switch pi-test:33861 [93] S ==> pi-test:33864 [97]
[33864] Task C starting
[33864] Task C grabbing lock
[33864] Task C has lock
279566.407926 [000] pi-test-33864 sched_switch pi-test:33864 [97] S ==> pi-test:33863 [120]
279566.407932 [000] pi-test-33863 sched_switch pi-test:33863 [95] S ==> pi-test:33862 [120]
279566.407937 [000] pi-test-33862 sched_waking comm=pi-test pid=33864 prio=97 target_cpu=000
[33862] Task A starting
279566.407940 [000] pi-test-33862 sched_waking comm=pi-test pid=33863 prio=95 target_cpu=000
[33862] Task A grabbing lock
279566.407944 [000] pi-test-33862 sched_pi_setprio comm=pi-test pid=33864 oldprio=97 newprio=94
279566.407947 [000] pi-test-33862 sched_switch pi-test:33862 [94] S ==> pi-test:33864 [94]
279567.359532 [000] pi-test-33864 sched_switch pi-test:33864 [94] R ==> kworker/0:2:31996 [120]
279567.407868 [000] <idle>-0 sched_switch swapper/0:0 [120] R ==> pi-test:33864 [94]
[33864] Task C releasing lock
279568.018263 [000] pi-test-33864 sched_pi_setprio comm=pi-test pid=33864 oldprio=94 newprio=97
279568.018266 [000] pi-test-33864 sched_waking comm=pi-test pid=33862 prio=94 target_cpu=000
279568.018268 [000] pi-test-33864 sched_switch pi-test:33864 [97] R ==> pi-test:33862 [94]
[33862] Task A has lock
[33862] Task A releasing lock
[33862] Task A released lock
[33862] Task A finished
[..]
```

I think that’s enough