Introduction of ftrace

David Chang
dchang@suse.com

What is ftrace?

- An internal tracer to find what is happening inside the kernel
- Ftrace was developed by Steven Rostedt
- From Ingo Molnar’s rt patch for latency-trace and Steven’s logdev
- Ftrace has been included in the kernel since v2.6.27
- Not just function!
- A generic tracing frame work for linux kernel

Overview of ftrace

- Trace functions within the kernel
- Event tracepoints
  - scheduler, interrupts, etc
- Call graphs trace
- Kernel stack size
- Latency tracing

How does ftrace work?

- Use gcc's profiler option: -pg
  - -pg Generate extra code to write profile information suitable for the analysis program

```
--- hello-without-pg     2017-08-01 11:11:57.235802889 +0800
+++ hello-with-pg   2017-08-01 11:11:40.547607202 +0800
@@ -13,6 +13,7 @@
     .cfi_offset 6, -16
     movq %rsp, %rbp
     .cfi_def_cfa_register 6
+    call mcount
     movl $.LC0, %edi
     call puts
     popq %rbp
```

- That function must be implemented in assembly because the call does not follow the normal C ABI

How does ftrace work?

- Add special mcount function call
  - Every function in the kernel call a special function "mcount()"
  - Except inline and a few special functions
- When CONFIG_DYNAMIC_FTRACE is configured
  - mcount is converted to a NOP at boot time
- When function tracer is enabled
  - convert the call-sites back into trace calls

Trace data of ftrace

- Per CPU ring buffer for holding data
- The newest data may overwrite the oldest data
- The size of the ring buffer is configurable
  - By echoing $SIZE > **buffer_size_kb**

```
linux-kyyb:/sys/kernel/debug/tracing # cat buffer_size_kb
1408
linux-kyyb:/sys/kernel/debug/tracing # cat buffer_total_size_kb
5632
```

The tracing directory and files

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



# Reference: linux/Documentation/trace/ftrace.txt
```

available_events

- A list of events that can be enabled in tracing

```
linux-kyyb:/sys/kernel/debug/tracing # cat available_events
xfs:xfs_attr_list_sf
xfs:xfs_attr_list_sf_all
xfs:xfs_attr_list_leaf
xfs:xfs_attr_list_leaf_end
xfs:xfs_attr_list_full
xfs:xfs_attr_list_add
xfs:xfs_attr_list_wrong_blk
[...]
linux-kyyb:/sys/kernel/debug/tracing # wc -l available_events
1782 available_events
```

available_filter_functions

- A list of available functions that you can add to **set_ftrace_filter** and **set_ftrace_notrace**

```
linux-kyyb:/sys/kernel/debug/tracing # cat available_filter_functions
run_init_process
try_to_run_init_process
do_one_initcall
match_dev_by_uuid
name_to_dev_t
rootfs_mount
rootfs_mount
calibration_delay_done
calibrate_delay
do_audit_syscall_entry
[...]
linux-kyyb:/sys/kernel/debug/tracing # wc -l available_filter_functions
41553 available_filter_functions
```

available_tracers

- List different types of tracers

```
linux-kyyb:/sys/kernel/debug/tracing # cat available_tracers
blk function_graph wakeup_dl wakeup_rt wakeup function nop
```

The tracers

- Nop
  - "trace nothing" tracer
- Function
  - Trace all kernel functions
- function_graph
  - Similar to the function tracer
  - It provides the ability to draw a graph of function calls similar to C code

The tracers

- blk
  - The block tracer used by the blktrace user application
- wakeup
  - Traces and records the max latency that it takes for the highest priority task to get scheduled after it has been woken up
  - Traces all tasks as an average developer would expect
- wakeup_rt
  - Traces and records the max latency that it takes for just RT tasks
- wakeup_dl
  - Traces and records the max latency that it takes for a SCHED_DEADLINE task to be woken

current_tracer

- set or display the current tracer
- Enable tracer by echoing the tracer name into **current_tracer**

```
linux-kyyb:/sys/kernel/debug/tracing # cat available_tracers
blk function_graph wakeup_dl wakeup_rt wakeup function nop

linux-kyyb:/sys/kernel/debug/tracing # cat current_tracer
nop
linux-kyyb:/sys/kernel/debug/tracing # echo function > current_tracer
linux-kyyb:/sys/kernel/debug/tracing # cat current_tracer
function
```

trace

- The output of the trace

```
linux-kyyb:/sys/kernel/debug/tracing # cat trace
# tracer: function
#
# entries-in-buffer/entries-written: 204929/3484829   #P:4
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
       gpg-agent-1478  [003] ....  5999.431925: current_kernel_time64 <-__audit_syscall_entry
       gpg-agent-1478  [003] ....  5999.431925: SyS_rt_sigaction <-entry_SYSCALL_64_fastpath
       gpg-agent-1478  [003] ....  5999.431925: __might_fault ←SyS_rt_sigaction
[...]
linux-kyyb:/sys/kernel/debug/tracing # echo > trace
```

trace_pipe

- Output is the same as the trace file but this file is meant to be streamed with live tracing

```
linux-kyyb:/sys/kernel/debug/tracing # cat trace_pipe
CPU:3 [LOST 33205 EVENTS]
            gcin-1424  [003] ...1  6314.691112: generic_permission <-__inode_permission
            gcin-1424  [003] ...1  6314.691112: get_cached_acl_rcu <-generic_permission
            gcin-1424  [003] ...1  6314.691112: in_group_p <-generic_permission
            gcin-1424  [003] ...1  6314.691112: groups_search <-generic_permission
            gcin-1424  [003] ...1  6314.691113: security_inode_permission <-link_path_walk
            gcin-1424  [003] ...1  6314.691113: walk_component <-link_path_walk
            gcin-1424  [003] ...1  6314.691113: lookup_fast <-walk_component
            gcin-1424  [003] ...1  6314.691113: __d_lookup_rcu ←lookup_fast
[...]
```

tracing_on

- sets or displays whether writing to the trace ring buffer
  - Disable tracer : 0
  - Enable tracer : 1

```
linux-kyyb:/sys/kernel/debug/tracing # cat tracing_on
0
linux-kyyb:/sys/kernel/debug/tracing # echo 1 > tracing_on
linux-kyyb:/sys/kernel/debug/tracing # cat tracing_on
1
```

Stop ftrace tracing

```
linux-kyyb:/sys/kernel/debug/tracing # echo nop > current_tracer
linux-kyyb:/sys/kernel/debug/tracing # cat trace
# tracer: nop
#
# entries-in-buffer/entries-written: 0/0   #P:4
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
```

Stop ftrace tracing

```
linux-kyyb:/sys/kernel/debug/tracing # echo 0 > tracing_on
linux-kyyb:/sys/kernel/debug/tracing # cat trace| head -20
# tracer: function
#
# entries-in-buffer/entries-written: 205037/1953012   #P:4
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
     kworker/1:2-3998  [001] d..1  5004.844253: __internal_add_timer <-internal_add_timer
     kworker/1:2-3998  [001] d..1  5004.844253: _raw_spin_unlock_irqrestore <-add_timer_on
     kworker/1:2-3998  [001] ....  5004.844253: mutex_unlock <-process_one_work
     kworker/1:2-3998  [001] ....  5004.844254: ___might_sleep <-process_one_work
     kworker/1:2-3998  [001] ....  5004.844254: _cond_resched <-process_one_work
     kworker/1:2-3998  [001] ....  5004.844254: rcu_all_qs <-process_one_work
     kworker/1:2-3998  [001] ....  5004.844254: _raw_spin_lock_irq <-process_one_work
     kworker/1:2-3998  [001] d..1  5004.844255: pwq_dec_nr_in_flight <-worker_thread
     kworker/1:2-3998  [001] d..1  5004.844255: worker_enter_idle <-worker_thread
```

set_ftrace_filter

- Set tracing of specified functions

```
linux-kyyb:/sys/kernel/debug/tracing # cat set_ftrace_filter
#### all functions enabled ####

linux-kyyb:/sys/kernel/debug/tracing # echo e1000e_set_rx_mode > set_ftrace_filter
linux-kyyb:/sys/kernel/debug/tracing # cat set_ftrace_filter
e1000e_set_rx_mode [e1000e]

linux-kyyb:/sys/kernel/debug/tracing # echo e1000e_setup_rx_resources >>
set_ftrace_filter

linux-kyyb:/sys/kernel/debug/tracing # cat set_ftrace_filter
e1000e_set_rx_mode [e1000e]
e1000e_setup_rx_resources [e1000e]
```

set_ftrace_notrace

- Opposite to set_ftrace_filter
- Overrides set_ftrace_filter

```
linux-kyyb:/sys/kernel/debug/tracing # echo '*lock*' > set_ftrace_notrace
linux-kyyb:/sys/kernel/debug/tracing # cat set_ftrace_notrace
xen_pte_unlock
update_persistent_clock
read_persistent_clock
set_task_blockstep
user_enable_block_step
prepare_threshold_block
get_block_address.isra.0
allocate_threshold_blocks
[...]
```

Glob matching

- <match>* : will match functions that begin with <match>
- *<match> : will match functions that end with <match>
- *<match>* : will match functions that have <match> in it
- <match1>*<match2> : will match functions that begin with <match1> and end with <match2>

- The wildcard (*) is also used by bash, so it's best to wrap the input with quotes

```
linux-kyyb:/sys/kernel/debug/tracing # echo set* > set_ftrace_filter
bash: echo: write error: Invalid argument
linux-kyyb:/sys/kernel/debug/tracing # echo 'set*' > set_ftrace_filter
```

set_ftrace_filter

```
linux-kyyb:/sys/kernel/debug/tracing # echo '*e1000e*' > set_ftrace_filter
linux-kyyb:/sys/kernel/debug/tracing # cat set_ftrace_filter
e1000e_get_laa_state_82571 [e1000e]
e1000e_set_laa_state_82571 [e1000e]
e1000e_write_protect_nvm_ich8lan [e1000e]
e1000e_set_kmrn_lock_loss_workaround_ich8lan [e1000e]
e1000e_gig_downshift_workaround_ich8lan [e1000e]
e1000e_igp3_phy_powerdown_workaround_ich8lan [e1000e]
e1000e_setup_led_generic [e1000e]
e1000e_get_bus_info_pcie [e1000e]
e1000e_init_rx_addrs [e1000e]
e1000e_rar_get_count_generic [e1000e]
e1000e_rar_set_generic [e1000e]
e1000e_update_mc_addr_list_generic [e1000e]
e1000e_clear_hw_cntrs_base [e1000e]
e1000e_setup_fiber_serdes_link [e1000e]
e1000e_config_collision_dist_generic [e1000e]
e1000e_set_fc_watermarks [e1000e]
e1000e_setup_link_generic [e1000e]
e1000e_force_mac_fc [e1000e]
[...]
```

To remove filter from set_ftrace_filter

```
linux-kyyb:/sys/kernel/debug/tracing # echo '!*e1000e*' > set_ftrace_filter

linux-kyyb:/sys/kernel/debug/tracing # cat set_ftrace_filter
#### all functions enabled ####

Or
linux-kyyb:/sys/kernel/debug/tracing # echo > set_ftrace_filter
```

set_graph_function : what a function does

```
linux-kyyb:/sys/kernel/debug/tracing # echo do_vfs_ioctl > set_graph_function
linux-kyyb:/sys/kernel/debug/tracing # cat trace
# tracer: function_graph
#
# CPU  DURATION                  FUNCTION CALLS
# |     |   |                     |   |   |   |
 2)               |  do_vfs_ioctl() {
 2)               |    tty_ioctl() {
 2)   0.128 us    |      tty_paranoia_check();
 2)               |      __tty_check_change() {
 2)   0.128 us    |        is_ignored();
 2)   1.220 us    |      }
 2)               |      __might_fault() {
 2)               |        __might_sleep() {
 2)   0.113 us    |          ___might_sleep();
 2)   0.922 us    |        }
 2)   1.674 us    |      }
 2)               |      find_vpid() {
 2)   0.145 us    |        find_pid_ns();
 2)   0.834 us    |      }
 2)   0.073 us    |      pid_task();
 2)   0.120 us    |      put_pid();
 2) + 10.049 us   |    }
 2) + 11.115 us   |  }
```

delay of function_graph

'$' - greater than 1 second
'@' - greater than 100 milisecond
'*' - greater than 10 milisecond
'#' - greater than 1000 microsecond
'!' - greater than 100 microsecond
'+' - greater than 10 microsecond
' ' - less than or equal to 10 microsecond

Output of function tracing

```
linux-kyyb:/sys/kernel/debug/tracing # echo function > current_tracer
linux-kyyb:/sys/kernel/debug/tracing # cat trace
# tracer: function
#
# entries-in-buffer/entries-written: 205008/245323   #P:4
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
            bash-6941  [001] ....  8766.115357: current_kernel_time64 <-__audit_syscall_entry
            bash-6941  [001] ....  8766.115357: SyS_rt_sigaction <-entry_SYSCALL_64_fastpath
            bash-6941  [001] ....  8766.115358: __might_fault <-SyS_rt_sigaction
            bash-6941  [001] ....  8766.115358: __might_sleep <-__might_fault
            bash-6941  [001] ....  8766.115358: ___might_sleep <-__might_fault
            bash-6941  [001] ....  8766.115358: do_sigaction <-SyS_rt_sigaction
            bash-6941  [001] ....  8766.115359: _raw_spin_lock_irq <-do_sigaction
            bash-6941  [001] ....  8766.115359: __might_fault <-SyS_rt_sigaction
            bash-6941  [001] ....  8766.115359: __might_sleep <-__might_fault
            bash-6941  [001] ....  8766.115359: ___might_sleep <-__might_fault
[...]
```

Output of function tracing

- irqs-off: 'd' interrupts are disabled. '.' otherwise.

  Note: If the architecture does not support a way to read the irq flags variable, an 'X' will always be printed here

- need-resched:

  'N' both TIF_NEED_RESCHED and PREEMPT_NEED_RESCHED is set,

  'n' only TIF_NEED_RESCHED is set,

  'p' only PREEMPT_NEED_RESCHED is set,

  '.' otherwise.

Function triggers for filter

- A command to perform when function is hit
- **<function>:<trigger>[:count]**
- **trigger**
  - **mod** : enables function filtering per module
  - **traceon/traceoff** : turn tracing on and off when the specified functions are hit
  - **snapshot** : will cause a snapshot when the function is hit
  - **enable_event/disable_event** : enable or disable a trace event
    - enable_event/disable_event:<system>:<event>
  - **dump** : it will dump the contents of the ftrace ring buffer to the console
  - **cpudump** : it will dump the contents of the ftrace ring buffer for the current CPU that executed the function that triggered the dump to the console

Remove triggers

- To remove trigger without count:

  echo **’!<function>:<trigger>’** > set_ftrace_filter

- To remove trigger with a count:

  echo **‘!<function>:<trigger>:0’** > set_ftrace_filter

Examples of function triggers

- Disable tracing when a schedule bug is hit the first 5 times
  # echo '__schedule_bug:traceoff:5' > set_ftrace_filter

- The removes the traceoff trigger for __schedule_bug that have a counter
  # echo '**!**__schedule_bug:traceoff:**0**' > set_ftrace_filter

Tracing a specific module

```
linux-kyyb:/sys/kernel/debug/tracing # cat set_ftrace_filter
#### all functions enabled ####
linux-kyyb:/sys/kernel/debug/tracing # echo '*:mod:e1000e' > set_ftrace_filter
linux-kyyb:/sys/kernel/debug/tracing # cat set_ftrace_filter | wc -l
401
linux-kyyb:/sys/kernel/debug/tracing # cat set_ftrace_filter |head -20
e1000_set_d0_lplu_state_82571 [e1000e]
e1000_check_mng_mode_82574 [e1000e]
e1000_write_nvm_82571 [e1000e]
e1000_put_hw_semaphore_82571 [e1000e]
e1000_put_hw_semaphore_82573 [e1000e]
e1000_clear_vfta_82571 [e1000e]
e1000_led_on_82574 [e1000e]
e1000_set_d3_lplu_state_82574 [e1000e]
e1000_set_d0_lplu_state_82574 [e1000e]
e1000_validate_nvm_checksum_82571 [e1000e]
e1000_get_hw_semaphore_82571 [e1000e]
e1000_release_nvm_82571 [e1000e]
e1000_acquire_nvm_82571 [e1000e]
e1000_read_mac_addr_82571 [e1000e]
e1000_setup_link_82571 [e1000e]
...
```

events

- Write 0/1 to enable/disable tracing of all events

```
linux-kyyb:/sys/kernel/debug/tracing # ls events
lock              fib             i2c          mce      power          signal    tlb
btrfs             filelock        i915         mei      printk         skb       udp
cfg80211          filemap         iommu        migrate  random         snd_pcm   v4l2
clk               ftrace          irq          module   ras            sock      vb2
compaction        gpio            irq_vectors  mpx      raw_syscalls   spi       vmscan
context_tracking  hda             kmem         napi     rcu            swiotlb   vsyscall
drm               hda_controller  kvm          net      regmap         syscalls  workqueue
enable            hda_intel       kvmmmu       nmi      rpm            task      writeback
exceptions        header_event    libata       oom      sched          thermal   xen
fence             header_page     mac80211     pagemap  scsi           timer     xfs
```

events/<system>

- Write 0/1 to enable/disable tracing of all systems
- If set filter, only events passing filter are traced

```
linux-kyyb:/sys/kernel/debug/tracing # ls events/sched
enable                  sched_process_exec  sched_stat_iowait   sched_wait_task
filter                  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
sched_pi_setprio        sched_stat_blocked  sched_switch
```

events/<system>/events

- enable - Write 0/1 to enable/disable tracing of <event>

```
linux-kyyb:/sys/kernel/debug/tracing # ls events/sched/sched_process_wait
enable  filter  format  id  trigger

linux-kyyb:/sys/kernel/debug/tracing # echo 1 > events/sched/sched_process_wait
linux-kyyb:/sys/kernel/debug/tracing # cat set_event
sched:sched_process_wait

Or
linux-kyyb:/sys/kernel/debug/tracing # echo sched_process_wait > set_event
```

events/<system>/events

- format – contains a description of each field in a logged event

```
linux-kyyb:/sys/kernel/debug/tracing # cat events/sched/sched_process_wait/format
name: sched_process_wait
ID: 268
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 comm[16];     offset:8; size:16;  signed:1;
     field:pid_t pid;    offset:24;     size:4;   signed:1;
     field:int prio;     offset:28;     size:4;   signed:1;

print fmt: "comm=%s pid=%d prio=%d", REC->comm, REC->pid, REC->prio
```

events/<system>/events

- filter - If set, only events passing filter are traced
- filter expressions syntax:
  - **Field-name relational-operator value**
    - field-names available can be found in the format
    - operators for numeric : ==, !=, <, <=, >, >=, &
    - operators for string : ==, !=, ~

events/<system>/events

- trigger - If set, a command to perform when event is hit
  - <triggers>[:count][**if <filter>**]

    triggers
    - traceon, traceoff, stacktrace, snapshot
    - enable_event:<system>:<event>
    - disable_event:<system>:<event>

  - To remove triggers
    - echo '**!**<trigger>' > <system>/<event>/trigger

Examples of event trigger

- **echo traceoff:3 > events/block/block_unplug/trigger**
  - If block_unplug trigger 3 time then trace of

- **echo 'enable_event:kmem:kmalloc:3 if nr_rq > 1' > events/block/block_unplug/trigger**
  - If block_unplug hit 3 times and if nr_rq >1, then enable kmem:kmalloc trace event

- **echo 'enable_event:net:net_dev_xmit if irq==27' > events/irq/irq_handler_entry/trigger**
  - If irq is 27 then enable net:net_dev_xmit trace event

trace_marker

- Synchronize between what is happening in user space and inside kernel
- A way to write into the ftrace kernel ring buffer from user space

```
linux-kyyb:/sys/kernel/debug/tracing # echo "Hello tracing" > trace_marker
linux-kyyb:/sys/kernel/debug/tracing # cat trace
# tracer: nop
#
# entries-in-buffer/entries-written: 1/1   #P:4
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
            bash-3375  [001] ...1  6975.839894: tracing_mark_write: Hello
tracing
```

trace_clock

- ftrace default uses the "local" clock. This clock is very fast and strictly per cpu.
  - **global**: This clock is in **sync with all CPUs** but may be a bit slower than the local clock.

```
linux-kyyb:/sys/kernel/debug/tracing # cat trace_clock
[local] global counter uptime perf mono mono_raw x86-tsc

linux-kyyb:/sys/kernel/debug/tracing # echo global > trace_clock

linux-kyyb:/sys/kernel/debug/tracing # cat trace_clock
local [global] counter uptime perf mono mono_raw x86-tsc
```

trace_options

- The trace_options file (or the options directory) is used to control what gets printed in the trace output, or manipulate the tracers.

```
linux-kyyb:/sys/kernel/debug/tracing # ls options/
annotate           funcgraph-cpu       function-trace   print-parent  test_nop_accept
bin                funcgraph-duration  graph-time       raw           test_nop_refuse
blk_classic        funcgraph-irqs      hex              record-cmd    trace_printk
block              funcgraph-overhead  irq-info         sleep-time    userstacktrace
context-info       funcgraph-overrun   latency-format   stacktrace    verbose
disable_on_free    funcgraph-proc      markers          sym-addr
display-graph      funcgraph-tail      overwrite        sym-offset
funcgraph-abstime  func_stack_trace    printk-msg-only  sym-userobj
```

trace_options

- To enable an option

  # echo sym-offset > trace_options

- To **disable** the options, echo in the option prepended with "no".

  # echo **no**print-parent > trace_options

trace-cmd

What is trace-cmd ?

- It’s a user-space front-end command-line tool for Ftrace
- It works with Ftrace
  - instead of echoing various commands into strange files
  - reading the result from another file

- # zypper in trace-cmd

trace-cmd

```
dchang@linux-kyyb:~> trace-cmd

trace-cmd version 2.5.1

usage:
  trace-cmd [COMMAND] ...

  commands:
     record - record a trace into a trace.dat file
     start - start tracing without recording into a file
     extract - extract a trace from the kernel
     stop - stop the kernel from recording trace data
     restart - restart the kernel trace data recording
     show - show the contents of the kernel tracing buffer
     reset - disable all kernel tracing and clear the trace buffers
     report - read out the trace stored in a trace.dat file
     stream - Start tracing and read the output directly
     profile - Start profiling and read the output directly
     hist - show a historgram of the trace.dat information
     stat - show the status of the running tracing (ftrace) system
     split - parse a trace.dat file into smaller file(s)
     options - list the plugin options available for trace-cmd report
     listen - listen on a network socket for trace clients
     list - list the available events, plugins or options
     restore - restore a crashed record
[...]
```

trace-cmd list

trace-cmd list

- list the available plugins or events that can be recorded

```
linux-kyyb:/home/dchang # trace-cmd list -e '^net*'
net:netif_rx_ni_entry
net:netif_rx_entry
net:netif_receive_skb_entry
net:napi_gro_receive_entry
net:napi_gro_frags_entry
[...]
linux-kyyb:/home/dchang # trace-cmd list -f
run_init_process
try_to_run_init_process
do_one_initcall
match_dev_by_uuid
calibration_delay_done
[...]
```

trace-cmd start / stop / show

trace-cmd start

- Uses same options as record, but does not run a command.
- It only enables the tracing and exits

trace-cmd stop

- Stops the tracer from recording more data.

trace-cmd show

- Basically, this is `cat trace`
- -p read the trace_pipe file instead, `cat trace_pipe`

Function tracing

```
linux-kyyb:/home/dchang # trace-cmd start -p function
  plugin 'function'
linux-kyyb:/home/dchang # trace-cmd show
# tracer: function
#
# entries-in-buffer/entries-written: 204987/9971752   #P:4
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
 gnome-terminal--2133  [000] d.h.  6430.959092: tick_program_event <-hrtimer_interrupt
 gnome-terminal--2133  [000] d.h.  6430.959093: clockevents_program_event <-hrtimer_interrupt
 gnome-terminal--2133  [000] d.h.  6430.959093: ktime_get <-clockevents_program_event
 gnome-terminal--2133  [000] d.h.  6430.959093: lapic_next_event <-clockevents_program_event
 gnome-terminal--2133  [000] d.h.  6430.959094: irq_exit <-smp_apic_timer_interrupt
 gnome-terminal--2133  [000] d...  6430.959094: __do_softirq <-irq_exit
 gnome-terminal--2133  [000] ..s.  6430.959094: run_timer_softirq <-__do_softirq
 gnome-terminal--2133  [000] ..s.  6430.959095: _raw_spin_lock_irq <-run_timer_softirq
 gnome-terminal--2133  [000] d.s.  6430.959095: call_timer_fn <-run_timer_softirq
 gnome-terminal--2133  [000] d.s.  6430.959095: delayed_work_timer_fn <-call_timer_fn
...
```

Function tracing (trace_pipe)

```
linux-kyyb:/home/dchang # trace-cmd show -p
CPU:3 [LOST 58725 EVENTS]
     gnome-shell-1529  [003] ....  7039.136925: fput <-do_sys_poll
     gnome-shell-1529  [003] ....  7039.136926: __fdget <-do_sys_poll
     gnome-shell-1529  [003] ....  7039.136926: __fget_light <-do_sys_poll
     gnome-shell-1529  [003] ....  7039.136926: __fget <-__fget_light
     gnome-shell-1529  [003] ....  7039.136926: timerfd_poll <-do_sys_poll
     gnome-shell-1529  [003] ....  7039.136926: _raw_spin_lock_irqsave <-timerfd_poll
     gnome-shell-1529  [003] d..1  7039.136926: _raw_spin_unlock_irqrestore <-timerfd_poll
     gnome-shell-1529  [003] ....  7039.136926: fput <-do_sys_poll
     gnome-shell-1529  [003] ....  7039.136926: __fdget <-do_sys_poll
     gnome-shell-1529  [003] ....  7039.136926: __fget_light <-do_sys_poll
     gnome-shell-1529  [003] ....  7039.136927: __fget <-__fget_light
     gnome-shell-1529  [003] ....  7039.136927: timerfd_poll <-do_sys_poll
     gnome-shell-1529  [003] ....  7039.136927: _raw_spin_lock_irqsave <-timerfd_poll
     gnome-shell-1529  [003] d..1  7039.136927: _raw_spin_unlock_irqrestore <-timerfd_poll
     gnome-shell-1529  [003] ....  7039.136927: fput <-do_sys_poll
     gnome-shell-1529  [003] ....  7039.136927: __fdget <-do_sys_poll
     gnome-shell-1529  [003] ....  7039.136927: __fget_light <-do_sys_poll
     gnome-shell-1529  [003] ....  7039.136927: __fget <-__fget_light
     gnome-shell-1529  [003] ....  7039.136927: eventfd_poll <-do_sys_poll
     gnome-shell-1529  [003] ....  7039.136928: fput <-do_sys_poll
...
```

Stop tracing (echo 0 > tracing_on)

```
linux-kyyb:/home/dchang # trace-cmd stop
linux-kyyb:/home/dchang # trace-cmd show
# tracer: function
#
# entries-in-buffer/entries-written: 204997/1347162   #P:4
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
     gnome-shell-1546  [003] ....   470.880365: __fget <-__fget_light
     gnome-shell-1546  [003] ....   470.880366: sock_poll <-do_sys_poll
     gnome-shell-1546  [003] ....   470.880366: unix_poll <-sock_poll
     gnome-shell-1546  [003] ....   470.880366: fput <-do_sys_poll
     gnome-shell-1546  [003] ....   470.880367: poll_freewait <-do_sys_poll
     gnome-shell-1546  [003] ....   470.880367: remove_wait_queue <-poll_freewait
     gnome-shell-1546  [003] ....   470.880367: _raw_spin_lock_irqsave <-remove_wait_queue
     gnome-shell-1546  [003] d..1   470.880367: _raw_spin_unlock_irqrestore <-poll_freewait
...
```

Stop tracing

```
linux-kyyb:/home/dchang # trace-cmd start -p nop
linux-kyyb:/home/dchang # trace-cmd show
# tracer: nop
#
# entries-in-buffer/entries-written: 0/0   #P:4
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
```

Function tracing filter

```
linux-kyyb:/home/dchang # trace-cmd start -p function -l ‘*e1000e*’
  plugin 'function'
linux-kyyb:/home/dchang # trace-cmd show
# tracer: function
#
# entries-in-buffer/entries-written: 51/51   #P:4
#
#                              _-----=> irqs-off
#                             / _----=> need-resched
#                            | / _---=> hardirq/softirq
#                            || / _--=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
     kworker/1:1-2245  [001] ....  2463.102738: e1000e_has_link <-e1000_watchdog_task
     kworker/1:1-2245  [001] ....  2463.102741: e1000e_phy_has_link_generic <-
e1000_check_for_copper_link_ich8lan
     kworker/1:1-2245  [001] ....  2463.102746: e1000e_read_phy_reg_mdic <-
__e1000_read_phy_reg_hv
     kworker/1:1-2245  [001] ....  2463.102746: e1000e_read_phy_reg_mdic.part.3 <-
__e1000_read_phy_reg_hv
     kworker/1:1-2245  [001] ....  2463.102853: e1000e_read_phy_reg_mdic <-
__e1000_read_phy_reg_hv
...
```

trace-cmd record / report / reset

trace-cmd record [command ...]

- record a trace into a **trace.dat** file
- -o data output file [default trace.dat]

trace-cmd report

- report - read out the trace stored in a **trace.dat** file
- -i input file [default trace.dat]

trace-cmd reset

- Disable all kernel tracing and clear the trace buffers

Record a trace

```
linux-kyyb:/home/dchang # trace-cmd record -e xfs ls
[...]
linux-kyyb:/home/dchang # trace-cm report
version = 6
CPU 1 is empty
cpus=4
 tracker-miner-f-1883  [000] 45835.335119: xfs_getattr:          dev 8:4 ino 0x7bb059
 tracker-miner-f-1883  [000] 45835.335196: xfs_getattr:          dev 8:4 ino 0x7bb059
       trace-cmd-22762 [002] 45835.335276: xfs_create:           dev 8:4 dp ino 0x43 name
trace.dat.cpu3
       trace-cmd-22762 [002] 45835.335288: xfs_log_reserve:      dev 8:4 type CREATE t_ocnt 2
t_cnt 2 t_curr_res 163284 t_unit_res 163284 t_flags XLOG_TIC_INITED|XLOG_TIC_PERM_RESERV
reserveq empty writeq empty grant_reserve_cycle 1371 grant_reserve_bytes 15252696
grant_write_cycle 1371 grant_write_bytes 15252696 curr_cycle 1371 curr_block 29778 tail_cycle
1371 tail_block 29716
       trace-cmd-22762 [002] 45835.335290: xfs_ilock:            dev 8:4 ino 0x43 flags
IOLOCK_EXCL|ILOCK_EXCL caller 0xffffffffa0b7d586s
       trace-cmd-22762 [002] 45835.335292: xfs_perag_get:        dev 8:4 agno 0 refcount 1494
caller 0xffffffffa0b5a5ads
       trace-cmd-22762 [002] 45835.335292: xfs_perag_put:        dev 8:4 agno 0 refcount 1493
caller 0xffffffffa0b5a71cs
       trace-cmd-22762 [002] 45835.335293: xfs_perag_get:        dev 8:4 agno 0 refcount 1494
caller 0xffffffffa0b5a89bs
...
```

Trace record with event and filter (irq latency)

```
linux-kyyb:/home/dchang # trace-cmd record -p function_graph -l do_IRQ -e
irq_handler_entry sleep 10
linux-kyyb:/home/dchang # trace-cmd report
version = 6
CPU 0 is empty
cpus=4
          <idle>-0     [002] 46283.127584: funcgraph_entry:                    |
__irqentry_text_start() {
          <idle>-0     [002] 46283.127591: irq_handler_entry:    irq=25 name=ahci[0000:00:1f.2]
          <idle>-0     [002] 46283.127618: funcgraph_exit:       + 31.363 us  |  }
          <idle>-0     [002] 46283.127990: funcgraph_entry:                    |
__irqentry_text_start() {
          <idle>-0     [002] 46283.127992: irq_handler_entry:    irq=25 name=ahci[0000:00:1f.2]
          <idle>-0     [002] 46283.128004: funcgraph_exit:       + 13.805 us  |  }
       trace-cmd-22968 [001] 46283.575566: funcgraph_entry:                    |
__irqentry_text_start() {
       trace-cmd-22968 [001] 46283.575626: irq_handler_entry:    irq=24 name=i915
       trace-cmd-22968 [001] 46283.575640: funcgraph_exit:       + 18.950 us  |  }
       trace-cmd-22968 [001] 46283.581302: funcgraph_entry:                    |
__irqentry_text_start() {
       trace-cmd-22968 [001] 46283.581302: irq_handler_entry:    irq=24 name=i915
       trace-cmd-22968 [001] 46283.581320: funcgraph_exit:       + 17.457 us  |  }
          <idle>-0     [003] 46283.763243: funcgraph_entry:                    |
...
```

trace-cmd report --events

- It will list the **event formats** of all events that were available in the created tracing file
- We can use the event which were available in the record
- To know what fields can be used for filtering a specific event

Report event format

```
linux-kyyb:/home/dchang # trace-cmd report –events | less
version = 6
name: wakeup
ID: 3
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 int prev_pid;    offset:8;       size:4; signed:0;
        field:unsigned int next_pid;    offset:12;      size:4; signed:0;
        field:unsigned int next_cpu;    offset:16;      size:4; signed:0;
        field:unsigned char prev_prio;  offset:20;      size:1; signed:0;
        field:unsigned char prev_state; offset:21;      size:1; signed:0;
        field:unsigned char next_prio;  offset:22;      size:1; signed:0;
        field:unsigned char next_state; offset:23;      size:1; signed:0;

print fmt: "%u:%u:%u  ==+ %u:%u:%u [%03u]", REC->prev_pid, REC->prev_prio, REC->prev_state, REC-
>next_pid, REC->next_prio, REC->next_state, REC->next_cpu
...
```

Example

- **trace-cmd start -v -e irq_handler_entry -R “enable_event:net:net_dev_xmit if irq==27”**

- Comparison:

  echo 'enable_event:net:net_dev_xmit if irq==27' > events/irq/irq_handler_entry/trigger

Tracing over network

- set up a trace server
  - $ trace-cmd listen -p 12345 -D -d /images/tracing/ -l /images/tracing/logfile

- Client
  - # trace-cmd record -N host:12345 -e sched_switch -e sched_wakeup -e irq hackbench 50

Q & A

bcc tools

- https://github.com/iovisor/bcc#tools

- tools/tcpaccept: Trace TCP passive connections (accept()) doesn't work, but tcptracer shows the events. Probably simple to fix.
- tools/tcpconnect: Trace TCP active connections (connect())
- tools/tcpconnlat: Trace TCP active connection latency (connect())
- tools/tcplife: Trace TCP sessions and summarize lifespan
- tools/tcpretrans: Trace TCP retransmits and TLPs
- tools/tcptop: Summarize TCP send/recv throughput by host. Top for TCP has a bug with rx_kb, submitted a fix
- tools/tcptracer: Trace TCP established connections (connect(), accept(), close())

Links

- Debugging the kernel using Ftrace - part 1
  - https://lwn.net/Articles/365835/
- Debugging the kernel using Ftrace - part 2
  - https://lwn.net/Articles/366796/
- Secrets of the Ftrace function tracer
  - https://lwn.net/Articles/370423/
- trace-cmd: A front-end for Ftrace
  - https://lwn.net/Articles/410200/
- https://www.kernel.org/doc/Documentation/trace/ftrace.txt
- https://www.kernel.org/doc/Documentation/trace/ftrace-design.txt
- https://www.kernel.org/doc/Documentation/trace/events.txt


SUSE
We adapt. You succeed.