Using the Linux Tracing Infrastructure and eBPF for Latency Analysis

Jan Altenberg
Open Source Automation Development Lab (OSADL) eG

whoami



AI generated content (Canva Business)



About OSADL

OSADL provides everything that is needed to **use, distribute and license FOSS components in industrial products**.

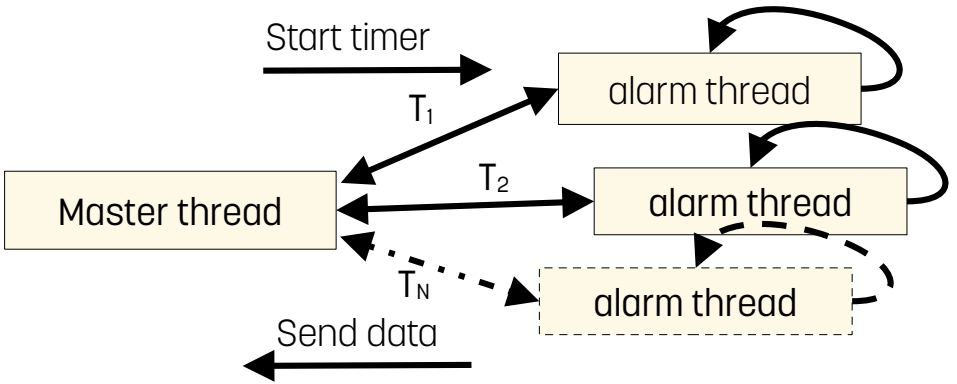
This includes the software itself, but also considers quality, licensing and compliance as well as safety and security aspects.

The OSADL business model is based on the principle:

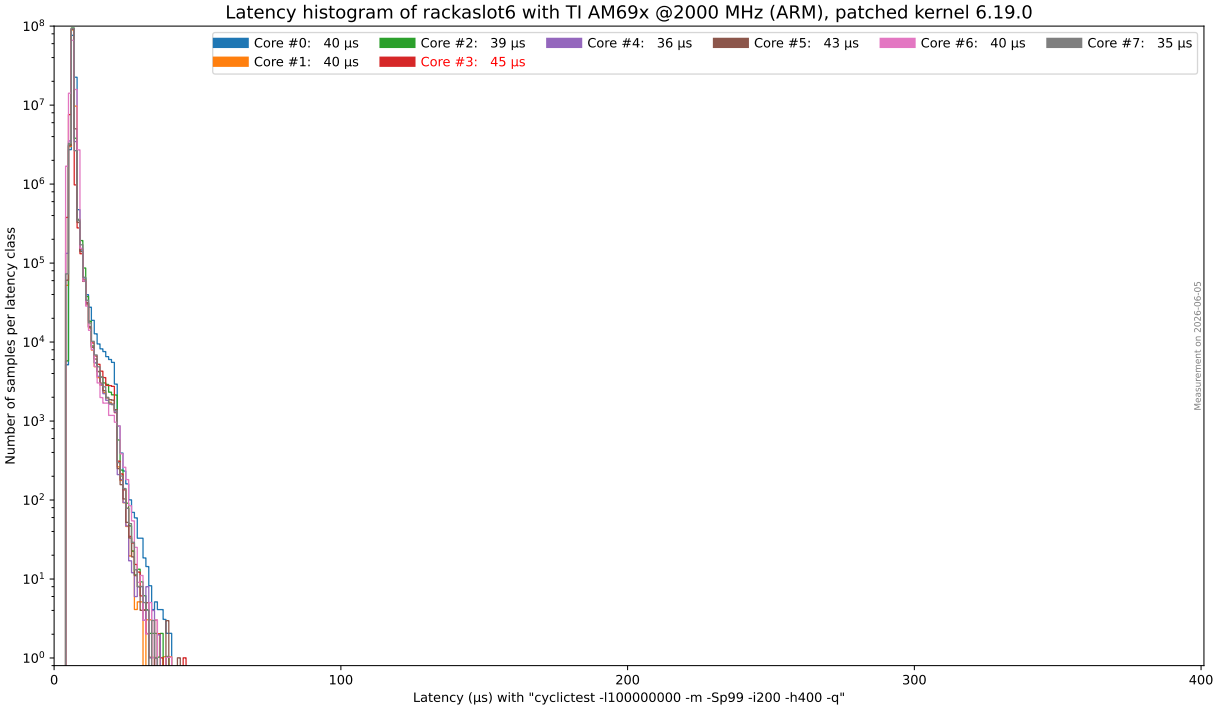
“Develop once for the use of many.”

cyclictest

Determination by a **simulated real-time application** (cyclictest):



Example: Latencyplot :-)

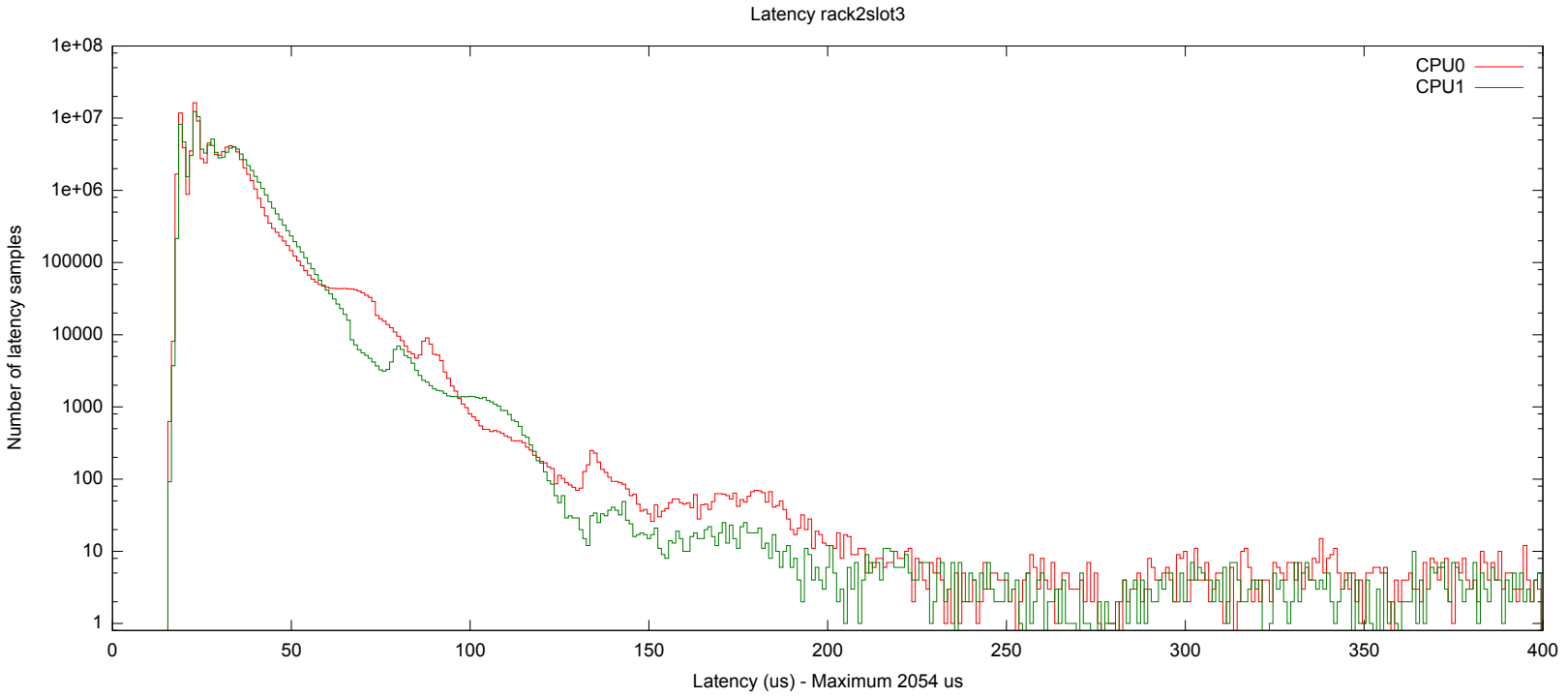


Example: Latencyplot :-)

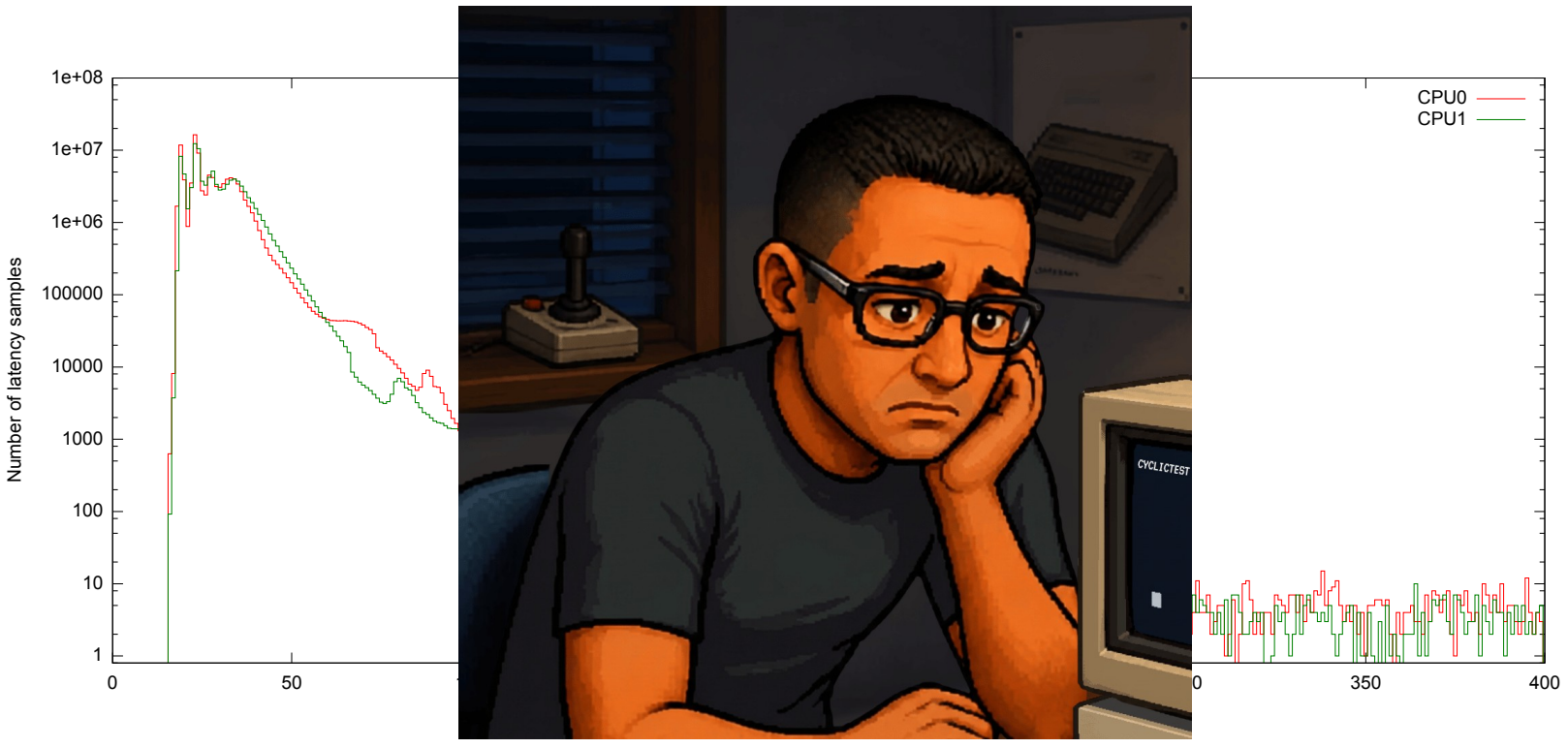


AI generated content (Canva Business)

Example: Latencyplot ;-(



Example: Latencyplot ;-(



AI generated content (Canva Business)

Problem

```
# cyclictest -m -S -i200 -d0 -p99
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 0.05 0.06 0.02 1/131 493

T: 0 (  474) P:99 I:200 C:  81314 Min:      6 Act:    9 Avg:    8 Max:    4952
T: 1 (  475) P:99 I:200 C:  81243 Min:      6 Act:    9 Avg:    8 Max:    4987
T: 2 (  476) P:99 I:200 C:  81191 Min:      6 Act:    9 Avg:    8 Max:    4880
T: 3 (  477) P:99 I:200 C:  81124 Min:      6 Act:   11 Avg:    8 Max:    4886
```

Problem

```
# cyclictest -m -S -i200 -d0 -p99
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 0.05 0.06 0.02 1/131 493

T: 0 (  474) P:99 I:200 C:  81314 Min:      6 Act:    9 Avg:    8 Max:    4952
T: 1 (  475) P:99 I:200 C:  81243 Min:      6 Act:    9 Avg:    8 Max:    4987
T: 2 (  476) P:99 I:200 C:  81191 Min:      6 Act:    9 Avg:    8 Max:    4880
T: 3 (  477) P:99 I:200 C:  81124 Min:      6 Act:   11 Avg:    8 Max:    4886
```

Let’s see what’s going wrong

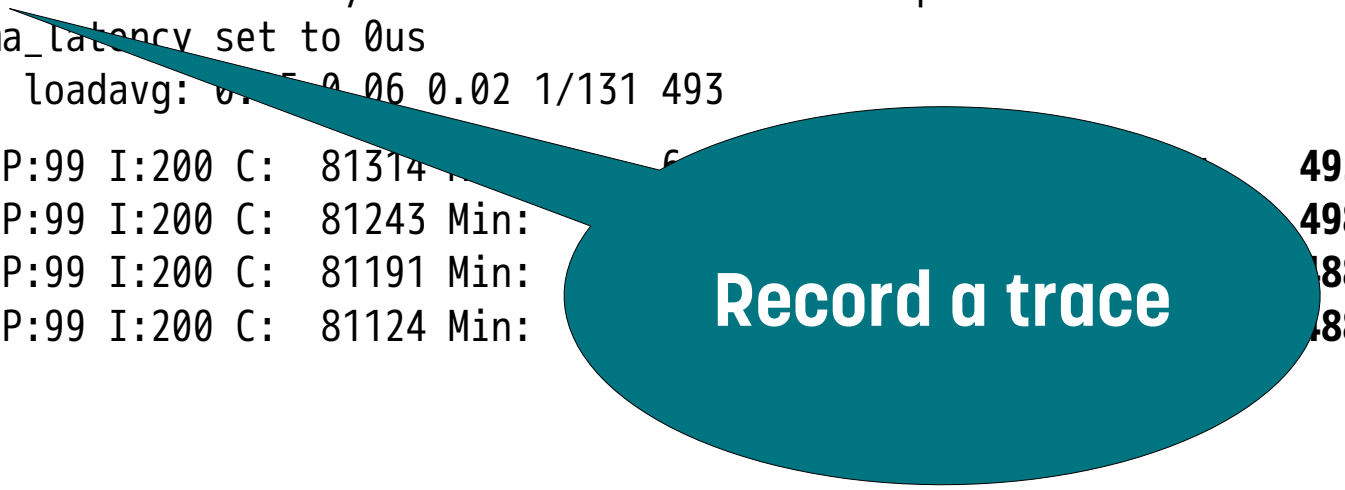
```
# trace-cmd record -e sched cyclictest -m -S -i200 -d0 -p99 -b4500 --tracemark
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 0.05 0.06 0.02 1/131 493

T: 0 (  474) P:99 I:200 C:  81314 Min:      6 Act:    9 Avg:    8 Max:    4952
T: 1 (  475) P:99 I:200 C:  81243 Min:      6 Act:    9 Avg:    8 Max:    4987
T: 2 (  476) P:99 I:200 C:  81191 Min:      6 Act:    9 Avg:    8 Max:    4880
T: 3 (  477) P:99 I:200 C:  81124 Min:      6 Act:   11 Avg:    8 Max:    4886
```

Let’s see what’s going wrong

```
# trace-cmd record -e sched cyclictest -m -S -i200 -d0 -p99 -b4500 --tracemark
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 0.[illegible] 0.06 0.02 1/131 493

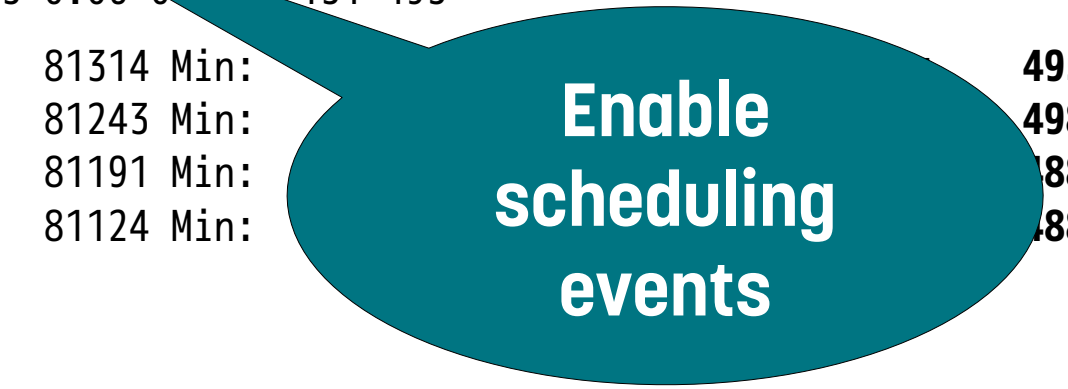
T: 0 (  474) P:99 I:200 C:  81314 [illegible]                                  4952
T: 1 (  475) P:99 I:200 C:  81243 Min:                                         4987
T: 2 (  476) P:99 I:200 C:  81191 Min:                                         4880
T: 3 (  477) P:99 I:200 C:  81124 Min:                                         4886
```



Let's see what's going wrong

```
# trace-cmd record -e sched cyclictest -m -S -i200 -d0 -p99 -b4500 --tracemark
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 0.05 0.06 0.[illegible] 1/131 493

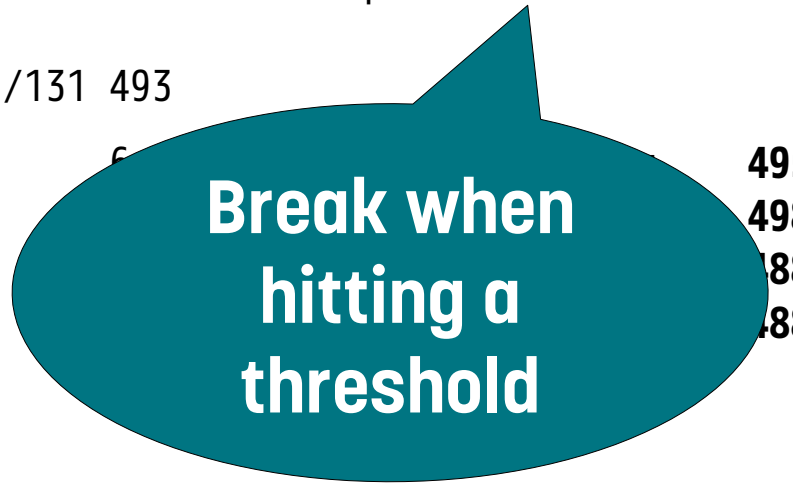
T: 0 (  474) P:99 I:200 C:  81314 Min:                                    4952
T: 1 (  475) P:99 I:200 C:  81243 Min:                                    4987
T: 2 (  476) P:99 I:200 C:  81191 Min:                                    4880
T: 3 (  477) P:99 I:200 C:  81124 Min:                                    4886
```



Let's see what's going wrong

```
# trace-cmd record -e sched cyclictest -m -S -i200 -d0 -p99 -b4500 --tracemark
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 0.05 0.06 0.02 1/131 493

T: 0 (  474) P:99 I:200 C:  81314 Min:                                  4952
T: 1 (  475) P:99 I:200 C:  81243 Min:                                  4987
T: 2 (  476) P:99 I:200 C:  81191 Min:                                  880
T: 3 (  477) P:99 I:200 C:  81124 Min:                                  886
```



Let's see what's going wrong

```
# trace-cmd record -e sched cyclictest -m -S -i200 -d0 -p99 -b4500 --tracemark
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 0.05 0.06 0.02 1/131 493

T: 0 (  474) P:99 I:200 C:  81314 Min:                                    4952
T: 1 (  475) P:99 I:200 C:  81243 Min:                                    4987
T: 2 (  476) P:99 I:200 C:  81191 Min:                                    4880
T: 3 (  477) P:99 I:200 C:  81124 Min:                                    4886
```



Let’s see what’s going wrong

```
# trace-cmd report

mklatency-771  [000] dNh3.  1766.250140: sched_waking: comm=irq/197-30be000 pid=109 prio=49 target_cpu=000
mklatency-771  [000] dNh4.  1766.250144: sched_wakeup: irq/197-30be000:109 [49] CPU:000
mklatency-771  [000] d..2.  1766.250151: sched_switch: mklatency:771 [120] R ==> cyclictest:766 [19]
cyclictest-766 [000] ...1.  1766.250165: print:        tracing_mark_write: hit latency threshold (4959 > 4500)
```

Let's see what's going wrong

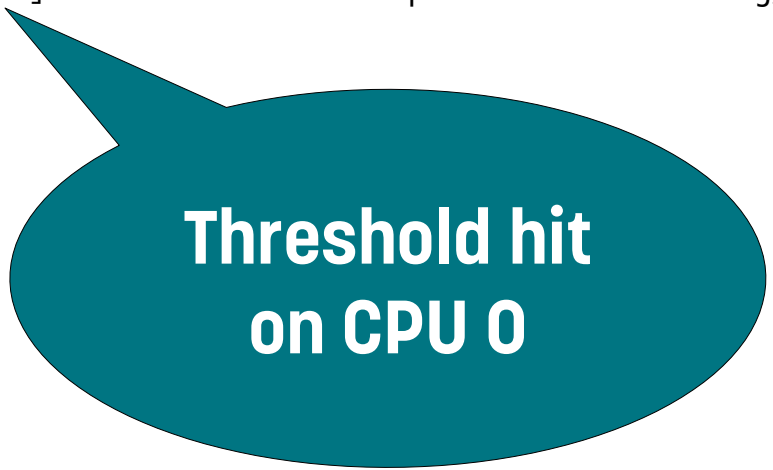
```
# trace-cmd report

mklatency-771  [000] dNh3.  1766.250140: sched_waking: comm=irq/197-30be000 pid=109 prio=49 target_cpu=000
mklatency-771  [000] dNh4.  1766.250144: sched_wakeup: irq/197-30be000:109 [49] CPU:000
mklatency-771  [000] d..2.  1766.250151: sched_switch: mklatency:771 [120] R ==> cyclictest:766 [19]
cyclictest-766 [000] ...1.  1766.250165: print:       tracing_mark_write: hit latency threshold (4959 > 4500)
```

Let's see what's going wrong

```
# trace-cmd report

mklatency-771  [000] dNh3.  1766.250140: sched_waking: comm=irq/197-30be000 pid=109 prio=49 target_cpu=000
mklatency-771  [000] dNh4.  1766.250144: sched_wakeup: irq/197-30be000:109 [49] CPU:000
mklatency-771  [000] d..2.  1766.250151: sched_switch: mklatency:771 [120] R ==> cyclictest:766 [19]
cyclictest-766 [000] ...1.  1766.250165: print:        tracing_mark_write: hit latency threshold (4959 > 4500)
```



Let's see what's going wrong

```
# trace-cmd report -cpu 0 --ts-diff

cyclictest-766   [000] d..2.  1766.245027: (+4)     sched_switch:       [...]
 mklatency-771   [000] d....  1766.245042: (+15)    blocksys_write:     [...]
 mklatency-771   [000] d.h3.  1766.245070: (+28)    sched_waking:       [...]
 mklatency-771   [000] dNh4.  1766.245074: (+4)     sched_wakeup:       [...]
 mklatency-771   [000] dN.2.  1766.245077: (+3)     sched_stat_runtime: [...]
 mklatency-771   [000] d..2.  1766.245081: (+4)     sched_switch:       [...]
irq_work/0-22    [000] d..2.  1766.245085: (+4)     sched_waking:       [...]
irq_work/0-22    [000] d..3.  1766.245092: (+7)     sched_wakeup:       [...]
irq_work/0-22    [000] d..2.  1766.245095: (+3)     sched_stat_runtime: [...]
irq_work/0-22    [000] d..2.  1766.245100: (+5)     sched_switch:       [...]
 mklatency-771   [000] d.h3.  1766.250105: (+5005)  sched_waking:       [...]
 mklatency-771   [000] dNh4.  1766.250108: (+3)     sched_wakeup:       [...]
```

Let's see what's going wrong

```
# trace-cmd report -cpu 0 --ts-diff

cyclictest-766   [000] d..2.  1766.245027: (+4)     sched_switch:       [...]
 mklatency-771   [000] d....                        sys_write:          [...]
 mklatency-771   [000] d.h3.                                            [...]
 mklatency-771   [000] dNh                                              [...]
 mklatency-771   [000] dN                                           me: [...]
 mklatency-771   [000] d..                                              [...]
irq_work/0-22    [000] d..2.                                       ng:  [...]
irq_work/0-22    [000] d..3.  1766.245092: (+7)     sched_wakeup:       [...]
irq_work/0-22    [000] d..2.  1766.245095: (+3)     sched_stat_runtime: [...]
irq_work/0-22    [000] d..2.  1766.245100: (+5)     sched_switch:       [...]
 mklatency-771   [000] d.h3.  1766.250105: (+5005)  sched_waking:       [...]
 mklatency-771   [000] dNh4.  1766.250108: (+3)     sched_wakeup:       [...]
```

Show events for CPU 0

Let's see what's going wrong

```
# trace-cmd report -cpu 0 --ts-diff

cyclictest-766   [000] d..2.  1766.[illegible]027: (+4)     sched_switch:       [...]
 mklatency-771   [000] d....  1766.[illegible]              [illegible]sys_write:     [...]
 mklatency-771   [000] d.h3.  [illegible]                   [illegible]:              [...]
 mklatency-771   [000] dNh[illegible]                                                 [...]
 mklatency-771   [000] dN[illegible]                                                  [...]
 mklatency-771   [000] d..[illegible]                                                 [...]
irq_work/0-22    [000] d..2.  [illegible]                   [illegible]ng:            [...]
irq_work/0-22    [000] d..3.  1766.245092: (+7)     sched_wakeup:       [...]
irq_work/0-22    [000] d..2.  1766.245095: (+3)     sched_stat_runtime: [...]
irq_work/0-22    [000] d..2.  1766.245100: (+5)     sched_switch:       [...]
 mklatency-771   [000] d.h3.  1766.250105: (+5005)  sched_waking:       [...]
 mklatency-771   [000] dNh4.  1766.250108: (+3)     sched_wakeup:       [...]
```

Show delta between timestamps

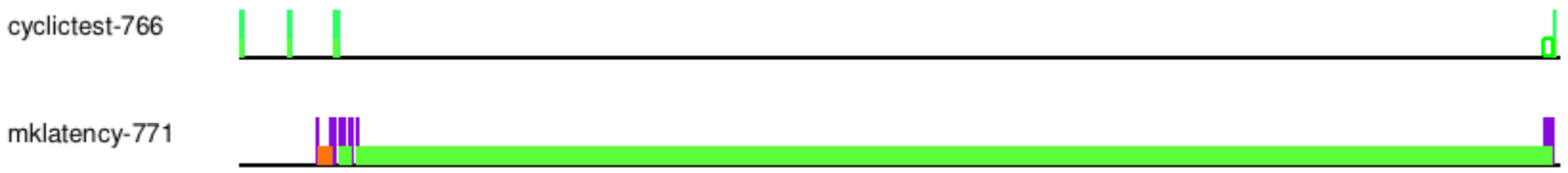
Let's see what's going wrong

```
# trace-cmd report -cpu 0 --ts-diff

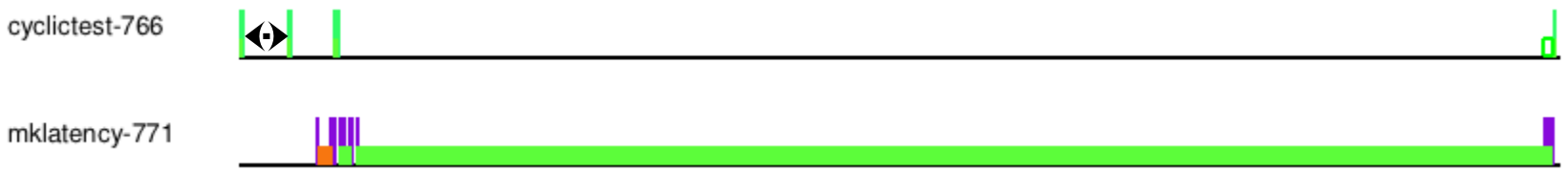
cyclictest-766   [000] d..2.  1766.[illegible]027: (+4)     sched_switch:       [...]
 mklatency-771   [000] d....  1766.[illegible]              [illegible]sys_write:     [...]
 mklatency-771   [000] d.h3.  [illegible]                   [illegible]:             [...]
 mklatency-771   [000] dNh[illegible]                                                [...]
 mklatency-771   [000] dN[illegible]                                                 [...]
 mklatency-771   [000] d..[illegible]                                                [...]
irq_work/0-22    [000] d..2.  [illegible]                   [illegible]ng:           [...]
irq_work/0-22    [000] d..3.  1766.245092: (+7)     sched_wakeup:       [...]
irq_work/0-22    [000] d..2.  1766.245095: (+3)     sched_stat_runtime: [...]
irq_work/0-22    [000] d..2.  1766.245100: (+5)     sched_switch:       [...]
 mklatency-771   [000] d.h3.  1766.250105: (+5005)  sched_waking:      [...]
 mklatency-771   [000] dNh4.  1766.250108: (+3)     sched_wakeup:       [...]
```

Show delta between timestamps

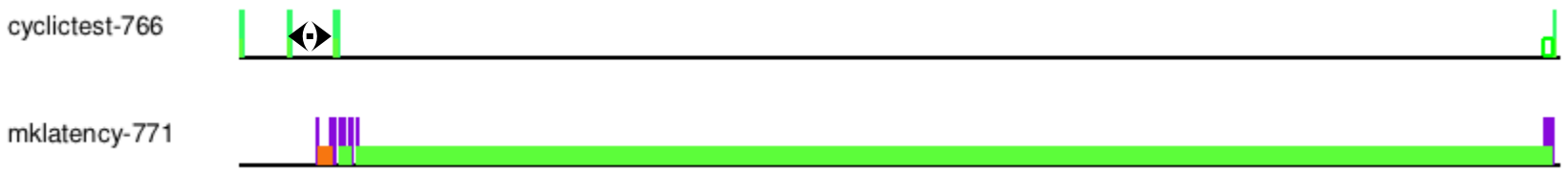
Let's see what's going wrong



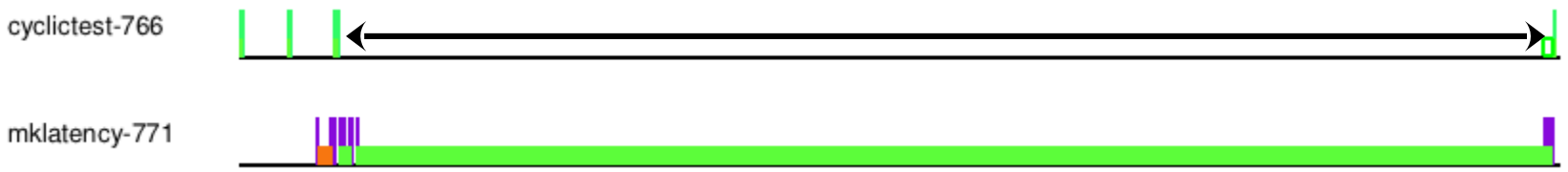
Let's see what's going wrong



Let's see what's going wrong



Let's see what's going wrong



Let’s see what’s going wrong



Let’s see what’s going wrong



Let’s see what’s going wrong



Let’s see what’s going wrong



Let’s see what’s going wrong

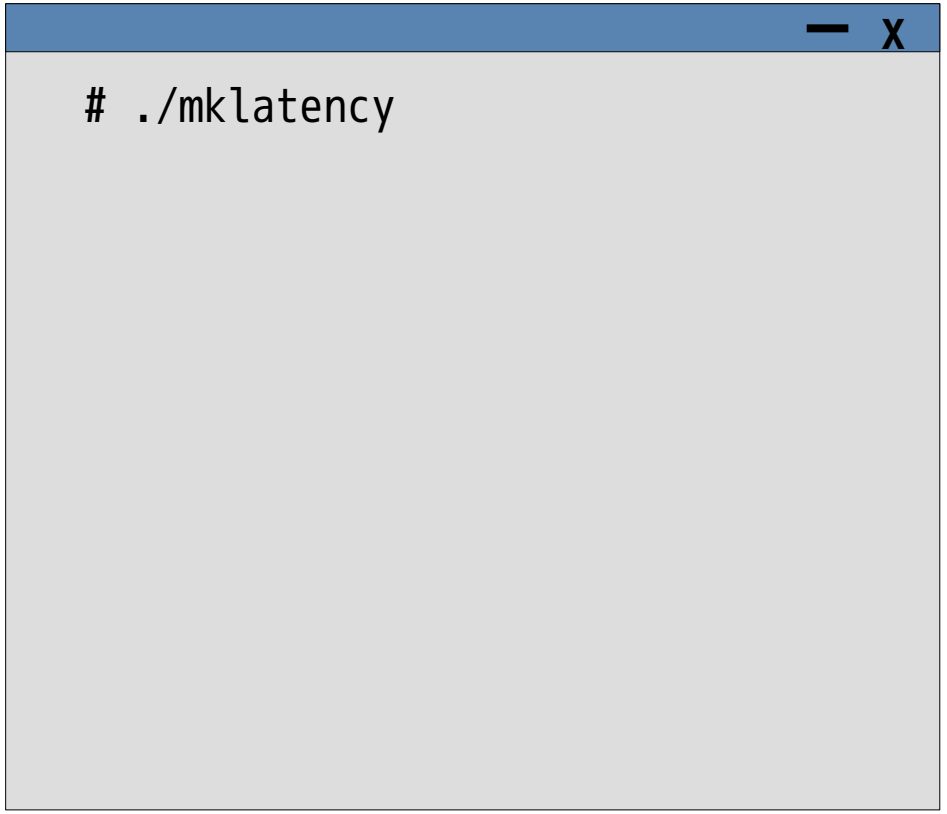


blocksys



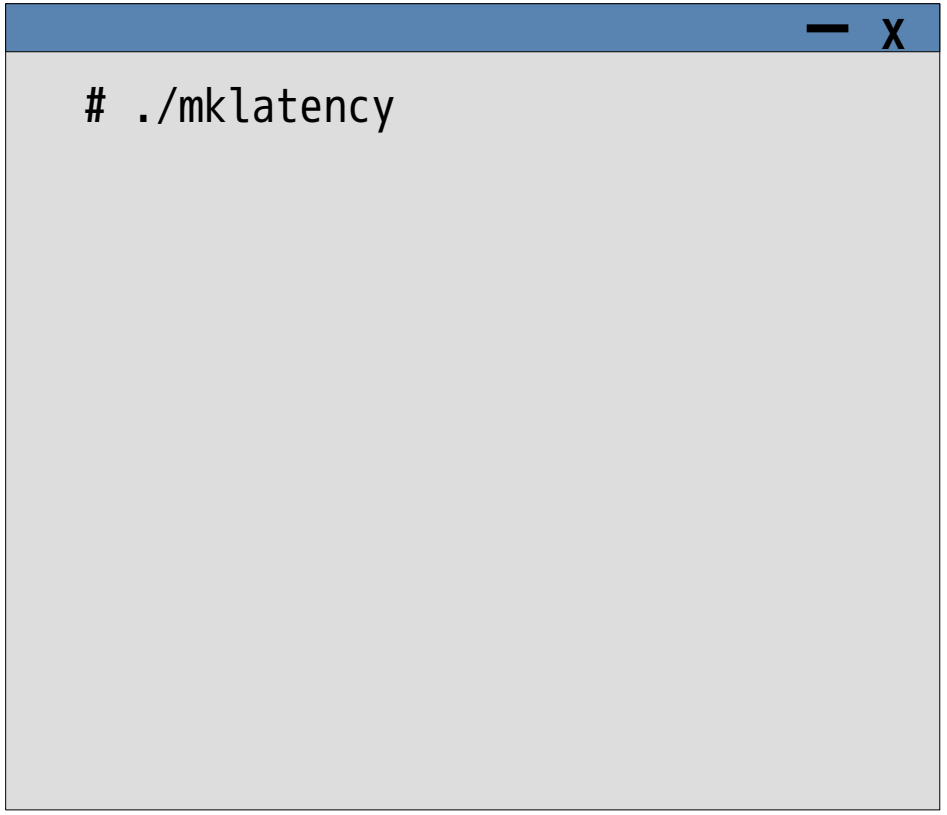

```
blocksys_write() {
[...]
  if (which == 'p' || which == 'b')
          preempt_disable();
  if (which == 'i' || which == 'b')
          local_irq_disable();
  while (nops--)
          asm("nop");
  if (which == 'i' || which == 'b')
          local_irq_enable();
  if (which == 'p' || which == 'b')
          preempt_enable();
[...]
}
```

blocksys




```
blocksys_write() {
[...]
  if (which == 'p' || which == 'b')
          preempt_disable();
  if (which == 'i' || which == 'b')
          local_irq_disable();
  while (nops--)
          asm("nop");
  if (which == 'i' || which == 'b')
          local_irq_enable();
  if (which == 'p' || which == 'b')
          preempt_enable();
[...]
}
```

blocksys




```
blocksys_write() {
[...]
  if (which == 'p' || which == 'b')
          preempt_disable();
  if (which == 'i' || which == 'b')
          local_irq_disable();
  while (nops--)
          asm("nop");
  if (which == 'i' || which == 'b')
          local_irq_enable();
  if (which == 'p' || which == 'b')
          preempt_enable();
[...]
}
```

blocksys

```
# ./mklatency
```

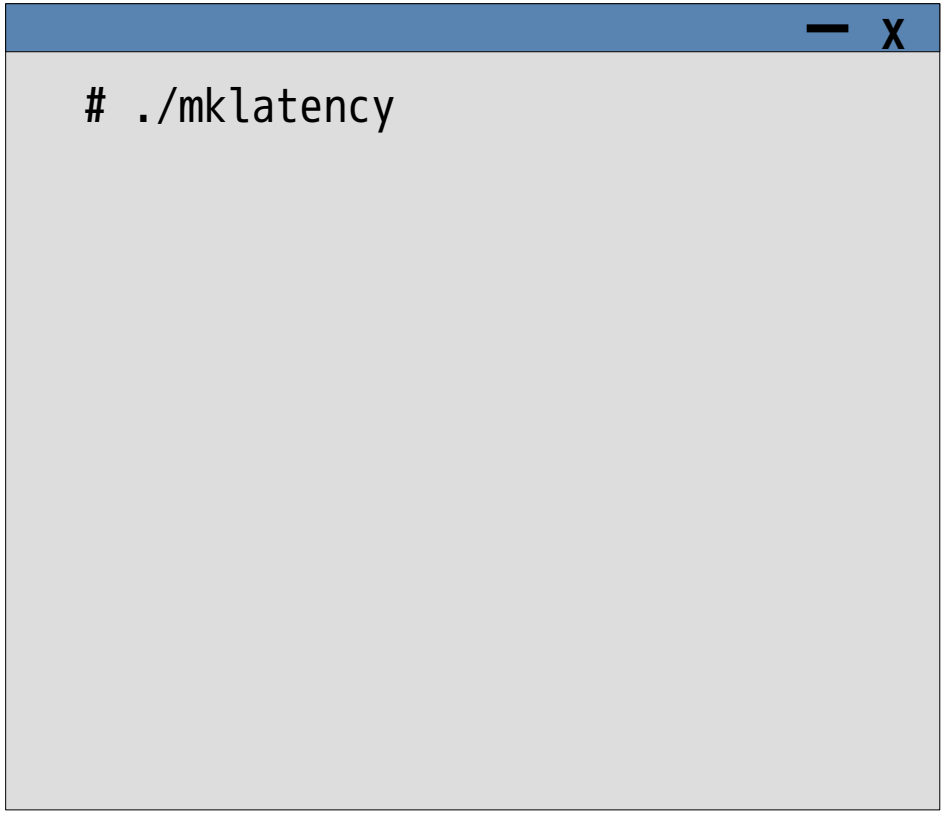
```
blocksys_write() {
[...]
  if (which == 'p' || which == 'b')
          preempt_disable();
  if (which == 'i' || which == 'b')
          local_irq_disable();
  while (nops--)
          asm("nop");
  if (which == 'i' || which == 'b')
          local_irq_enable();
  if (which == 'p' || which == 'b')
          preempt_enable();
[...]
}
```

blocksys

```
# ./mklatency
```

```
blocksys_write() {
[...]
  if (which == 'p' || which == 'b')
          preempt_disable();
  if (which == 'i' || which == 'b')
          local_irq_disable();
  while (nops--)
          asm("nop");
  if (which == 'i' || which == 'b')
          local_irq_enable();
  if (which == 'p' || which == 'b')
          preempt_enable();
[...]
}
```

blocksys




```
blocksys_write() {
[...]
  if (which == 'p' || which == 'b')
          preempt_disable();
  if (which == 'i' || which == 'b')
          local_irq_disable();
  while (nops--)
          asm("nop");
  if (which == 'i' || which == 'b')
          local_irq_enable();
  if (which == 'p' || which == 'b')
          preempt_enable();
[...]
}
```

Latency histograms patch

https://www.osadl.org/?id=2945

Latency histograms

```
# ls /sys/kernel/debug/latency_hist/
enable   missed_timer_offsets  switchtime  timerandwakeup
timerwakeupswitch  wakeup
```

Latency histograms

```
# cyclictest -m -S -i200 -d0 -p99
# /dev/cpu_dma_latency set to 0us
policy: fifo: loadavg: 0.05 0.06 0.02 1/131 493

T: 0 (  474) P:99 I:200 C:  81314 Min:      6 Act:    9 Avg:    8 Max:    4952
T: 1 (  475) P:99 I:200 C:  81243 Min:      6 Act:    9 Avg:    8 Max:    4987
T: 2 (  476) P:99 I:200 C:  81191 Min:      6 Act:    9 Avg:    8 Max:    4880
T: 3 (  477) P:99 I:200 C:  81124 Min:      6 Act:   11 Avg:    8 Max:    4886
```

Latency histograms

```
# cd /sys/kernel/debug/latency_hist/timerandwakeup

# cat max_latency-CPU*

546 99 4988 (4964,0) cyclictest <- 552 -21 mklatency 440.676730
547 99 4875 (4849,0) cyclictest <- 553 -21 mklatency 440.687647
548 99 4929 (4909,0) cyclictest <- 554 -21 mklatency 440.697379
549 99 4874 (4849,0) cyclictest <- 555 -21 mklatency 440.707655
```

Latency histograms

```
# cat max_latency-CPU*

546 99 4988 (4964,0) cyclictest <- 552 -21 mklatency 440.676730
547 99 4875 (4849,0) cyclictest <- 553 -21 mklatency 440.687647
548 99 4929 (4909,0) cyclictest <- 554 -21 mklatency 440.697379
549 99 4874 (4849,0) cyclictest <- 555 -21 mklatency 440.707655
```



Latency histograms

```
# cat max_latency-CPU*

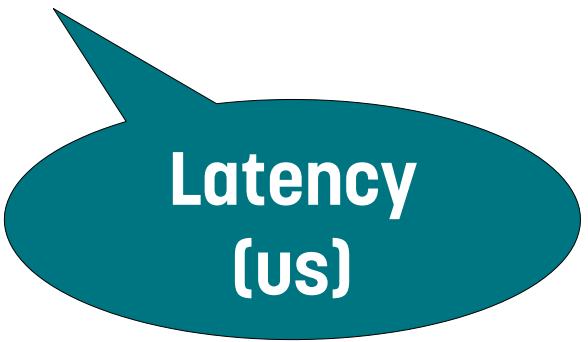
546 99 4988 (4964,0) cyclictest <- 552 -21 mklatency 440.676730
547 99 4875 (4849,0) cyclictest <- 553 -21 mklatency 440.687647
548 99 4929 (4909,0) cyclictest <- 554 -21 mklatency 440.697379
549 99 4874 (4849,0) cyclictest <- 555 -21 mklatency 440.707655
```



Latency histograms

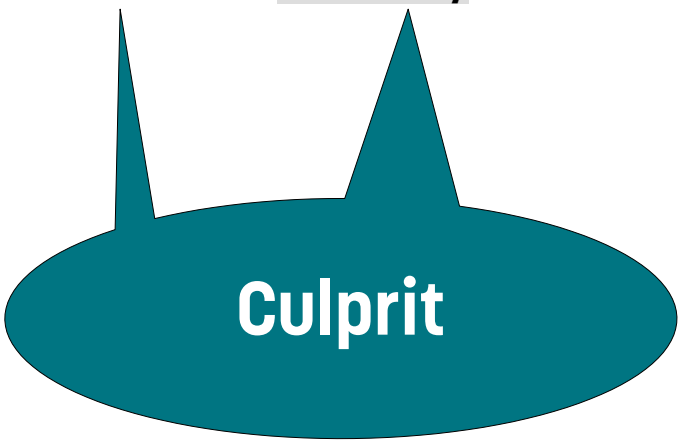
```
# cat max_latency-CPU*

546 99 4988 (4964,0) cyclictest <- 552 -21 mklatency 440.676730
547 99 4875 (4849,0) cyclictest <- 553 -21 mklatency 440.687647
548 99 4929 (4909,0) cyclictest <- 554 -21 mklatency 440.697379
549 99 4874 (4849,0) cyclictest <- 555 -21 mklatency 440.707655
```



Latency histograms

```
# cat max_latency-CPU*
546 99 4988 (4964,0) cyclictest <- 552 -21 mklatency 440.676730
547 99 4875 (4849,0) cyclictest <- 553 -21 mklatency 440.687647
548 99 4929 (4909,0) cyclictest <- 554 -21 mklatency 440.697379
549 99 4874 (4849,0) cyclictest <- 555 -21 mklatency 440.707655
```



Latency histograms

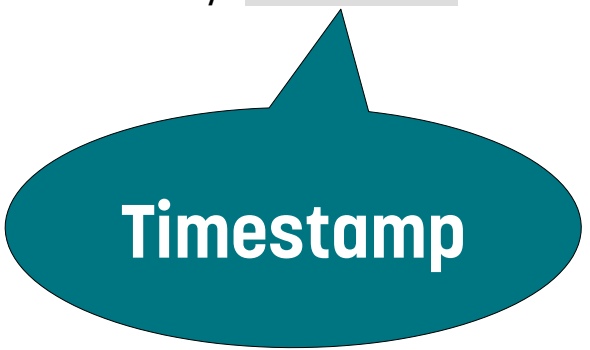
```
# cat max_latency-CPU*
546 99 4988 (4964,0) cyclictest <- 552 -21 mklatency 440.676730
547 99 4875 (4849,0) cyclictest <- 553 -21 mklatency 440.687647
548 99 4929 (4909,0) cyclictest <- 554 -21 mklatency 440.697379
549 99 4874 (4849,0) cyclictest <- 555 -21 mklatency 440.707655
```



Latency histograms

```
# cat max_latency-CPU*

546 99 4988 (4964,0) cyclictest <- 552 -21 mklatency 440.676730
547 99 4875 (4849,0) cyclictest <- 553 -21 mklatency 440.687647
548 99 4929 (4909,0) cyclictest <- 554 -21 mklatency 440.697379
549 99 4874 (4849,0) cyclictest <- 555 -21 mklatency 440.707655
```



RTLA: timerlat top

Timer Latency												
0 00:00:26	IRQ Timer Latency (us)				Thread Timer Latency (us)				Ret user Timer Latency (us)			
CPU COUNT	cur	min	avg	max	cur	min	avg	max	cur	min	avg	max
0 #26380	1	1	1	14	12	8	11	26	19	13	16	54
1 #26378	1	1	1	13	11	8	11	28	16	13	16	33
2 #26377	1	0	1	16	12	8	11	34	17	13	16	44
3 #26375	1	0	1	17	25	8	19	45	30	27	32	52
ALL #105510 e0		0	1	17		8	13	45		13	20	54

RTLA: timerlat top

Timer Latency												
0 00:00:26	IRQ Timer Latency (us)				Thread Timer Latency (us)				Ret user Timer Latency (us)			
CPU COUNT	cur	min	avg	max	cur	min	avg	max	cur	min	avg	max
0 #26380	1	1	1	14	12	8	11	26	19	13	16	54
1 #26378	1	1	1	13	11	8	11	28	16	13	16	33
2 #26377	1	0	1	16	12	8	11	34	17	13	16	44
3 #26375	1	0	1	17	25	8	19	45	30	27	32	52
ALL #105510 e0		0		17		8	13	45		13	20	54

Timer IRQ latency

RTLA: timerlat top

Timer Latency												
0 00:00:26	IRQ Timer Latency (us)				Thread Timer Latency (us)				Ret user Timer Latency (us)			
CPU COUNT	cur	min	avg	max	cur	min	avg	max	cur	min	avg	max
0 #26380	1	1	1	14	12	8	11	26	19	13	16	54
1 #26378	1	1	1	13	11	8	11	28	16	13	16	33
2 #26377	1	0	1	16	12	8	11	34	17	13	16	44
3 #26375	1	0	1	17	25	8	19	45	30	27	32	52
ALL #105510 e0		0	1	17			13	45		13	20	54

Timer latency kernel thread

RTLA: timerlat top

Timer Latency												
0 00:00:26	IRQ Timer Latency (us)				Thread Timer Latency (us)				Ret user Timer Latency (us)			
CPU COUNT	cur	min	avg	max	cur	min	avg	max	cur	min	avg	max
0 #26380	1	1	1	14	12	8	11	26	19	13	16	54
1 #26378	1	1	1	13	11	8	11	28	16	13	16	33
2 #26377	1	0	1	16	12	8	11	34	17	13	16	44
3 #26375	1	0	1	17	25	8	19	45	30	[illegible]	32	52
ALL #105510 e0		0	1	17		8	13	45		13	20	54

Timer latency user thread

RTLA: timerlat

```
# rtla timerlat top -a 4000 -Pf:99

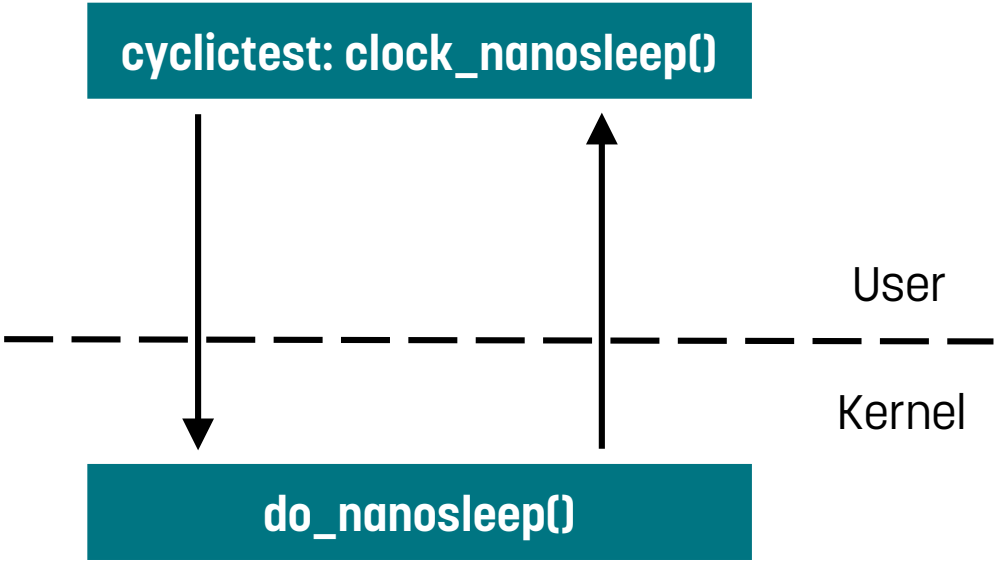
[...]

rtla timerlat hit stop tracing
## CPU 0 hit stop tracing, analyzing it ##

  IRQ handler delay:                                    0.00 us (0.00 %)
  IRQ latency:                                       4394.54 us
  [...]

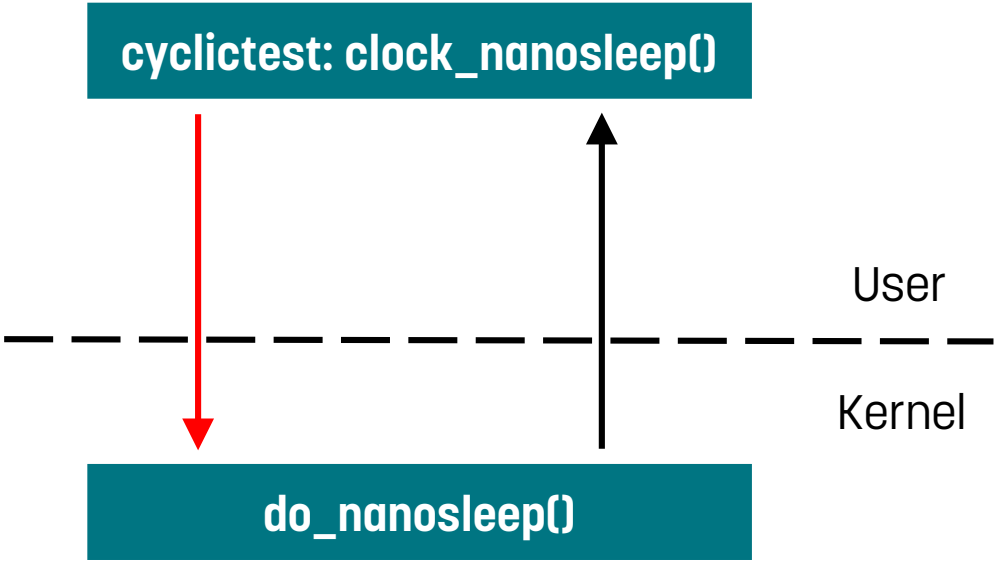
  Blocking thread stack trace
                -> timerlat_irq
                [...]
                -> blocksys_write
```

bpftrace: clock_nanosleep()



bpftrace: clock_nanosleep()

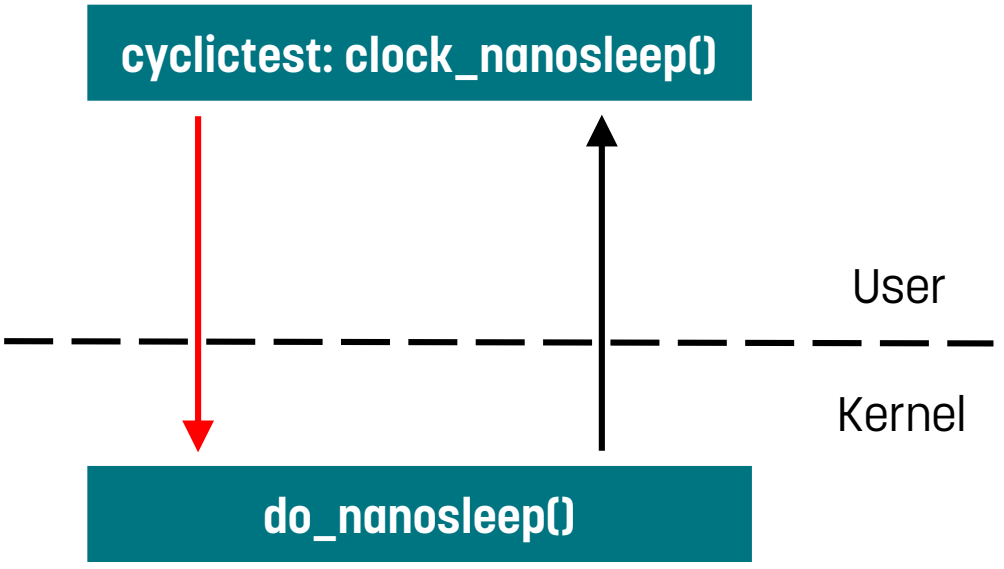
```
kprobe:do_nanosleep
```



bpftrace: clock_nanosleep()

```
kprobe:do_nanosleep
/ comm == "cyclictest" /
```

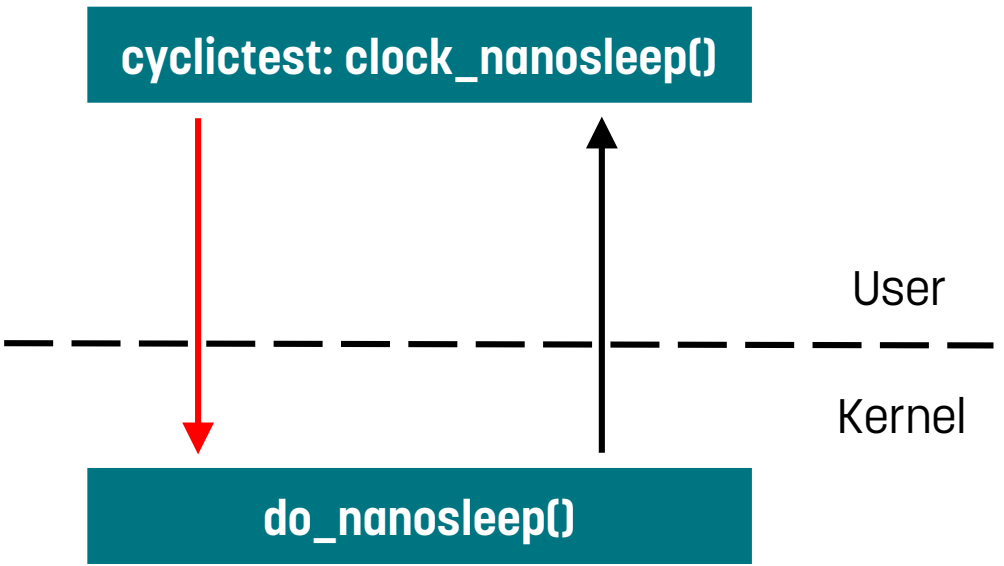
We are only interested in cyclictest



bpftrace: clock_nanosleep()

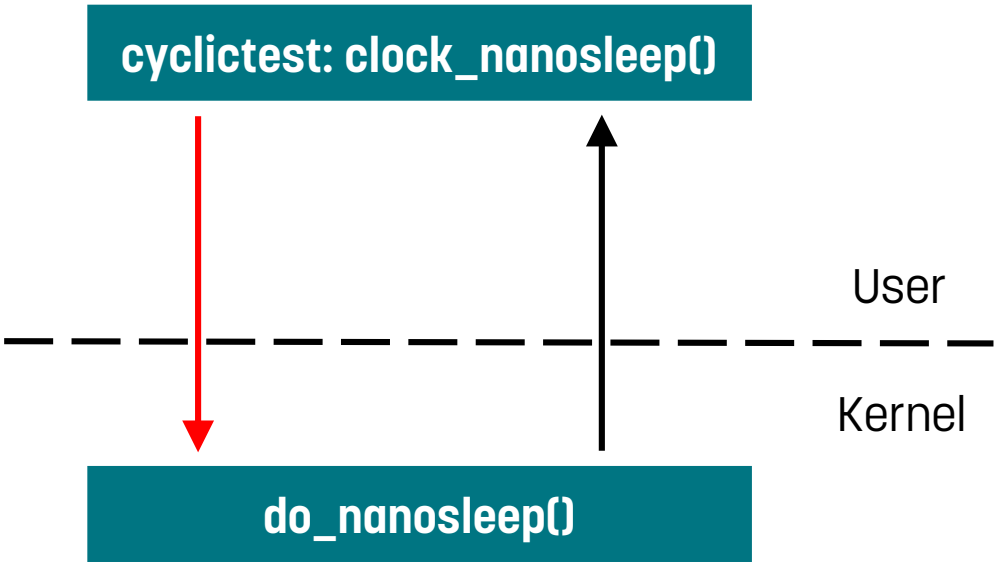
```
kprobe:do_nanosleep
/ comm == "cyclictest" && curtask->prio <= 99 /
```

We are only interested in threads with RT priority



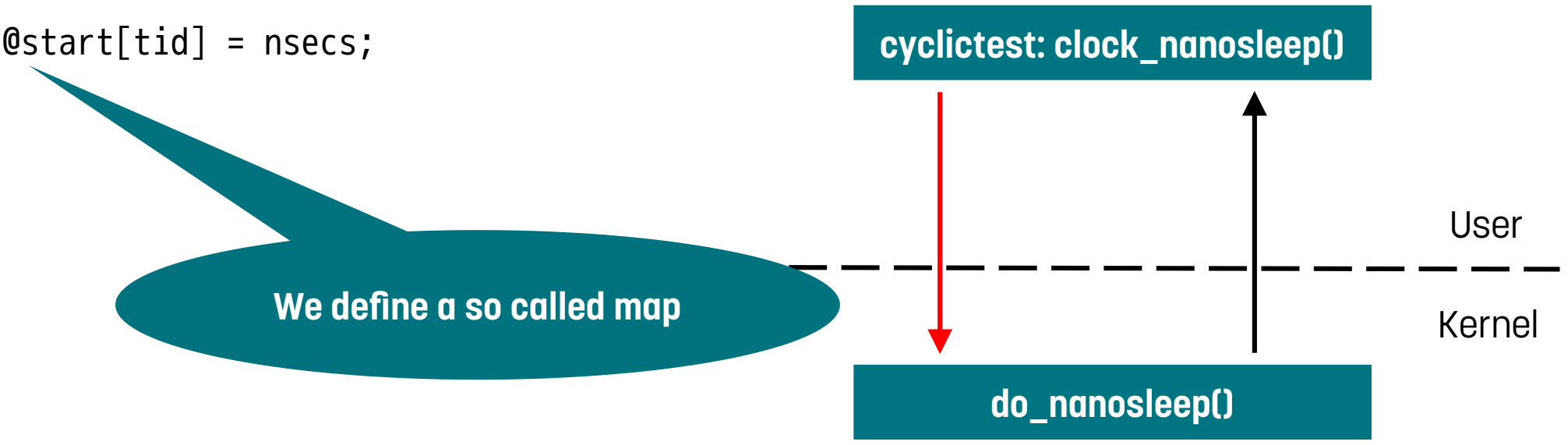
bpftrace: clock_nanosleep()

```
kprobe:do_nanosleep
/ comm == "cyclictest" && curtask->prio <= 99 /
{
  @start[tid] = nsecs;
}
```



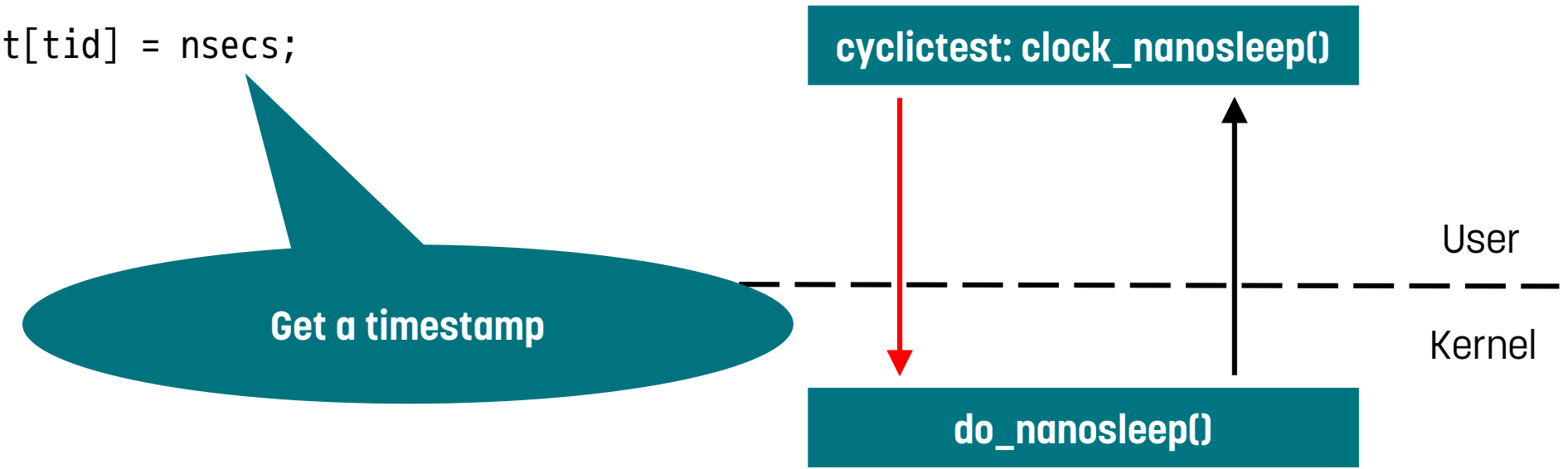
bpftrace: clock_nanosleep()

```
kprobe:do_nanosleep
/ comm == "cyclictest" && curtask->prio <= 99 /
{
  @start[tid] = nsecs;
}
```



bpftrace: clock_nanosleep()

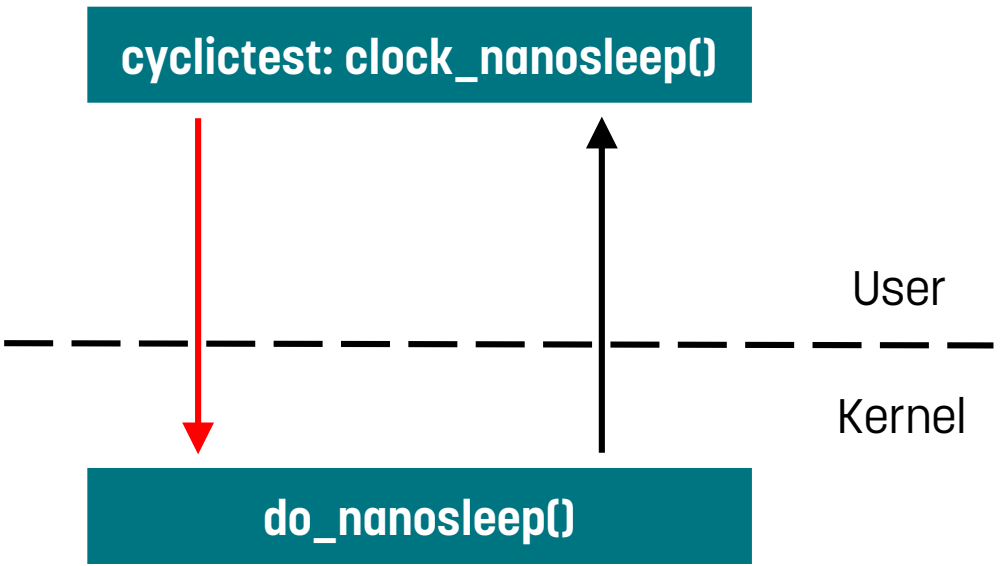
```
kprobe:do_nanosleep
/ comm == "cyclictest" && curtask->prio <= 99 /
{
  @start[tid] = nsecs;
}
```



bpftrace: clock_nanosleep()

```
kprobe:do_nanosleep
/ comm == "cyclictest" && curtask->prio <= 99 /
{
  @start[tid] = nsecs;
}
```

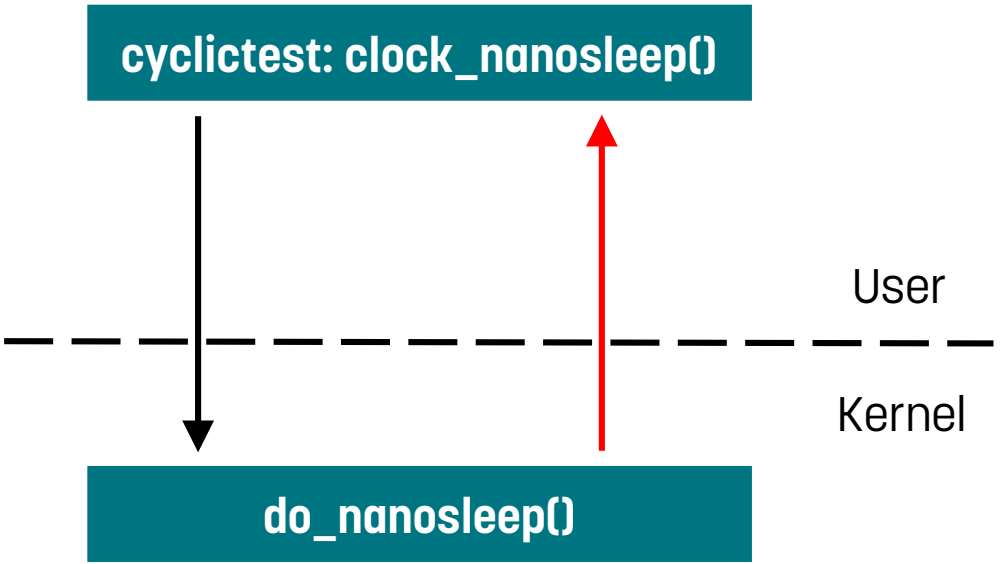
Save it based on the TID
(to avoid concurrency issues)



bpftrace: clock_nanosleep()

```
kprobe:do_nanosleep
/ comm == "cyclictest" && curtask->prio <= 99 /
{
  @start[tid] = nsecs;
}

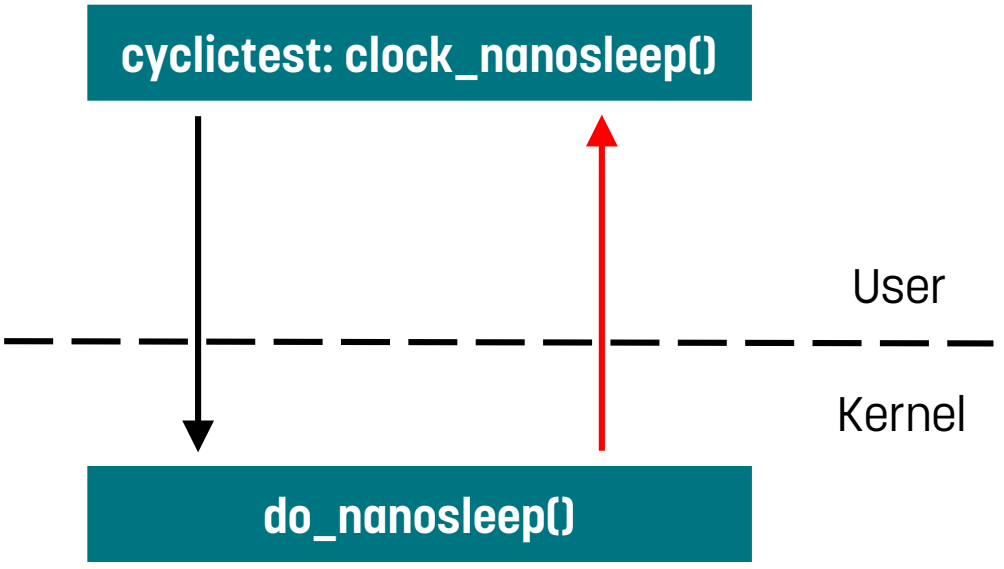
kretprobe:do_nanosleep
```



bpftrace: clock_nanosleep()

```
kprobe:do_nanosleep
/ comm == "cyclictest" && curtask->prio <= 99 /
{
  @start[tid] = nsecs;
}

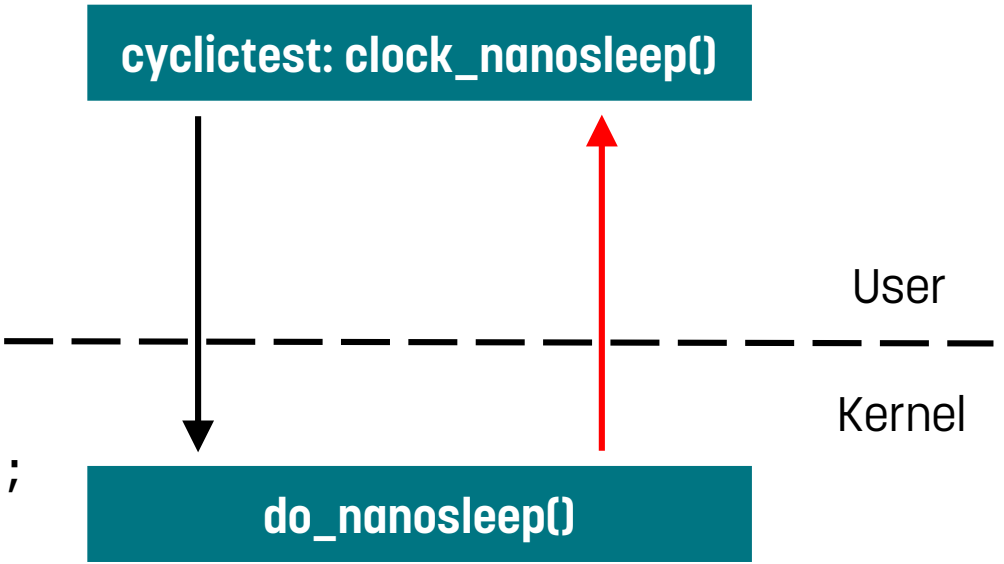
kretprobe:do_nanosleep
/ comm == "cyclictest" && @start[tid]
     != 0 && curtask->prio <= 99  /
```



bpftrace: clock_nanosleep()

```
kprobe:do_nanosleep
/ comm == "cyclictest" && curtask->prio <= 99 /
{
  @start[tid] = nsecs;
}

kretprobe:do_nanosleep
/ comm == "cyclictest" && @start[tid]
     != 0 && curtask->prio <= 99  /
{
  $delta = nsecs - @start[tid];
  printf("nanosleep() - %d nsecs\n", $delta);
  delete(@start[tid]);
}
```

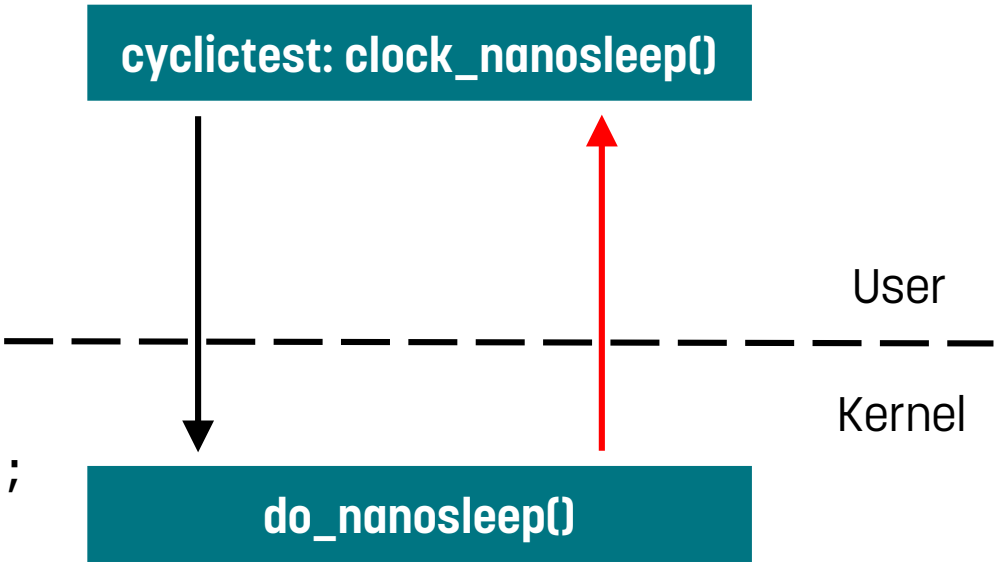


bpftrace: clock_nanosleep()

```
kprobe:do_nanosleep
/ comm == "cyclictest" && curtask->prio <= 99 /
{
  @start[tid] = nsecs;
}

kretpr
/ comm == "cyc                 start[tid]
     != 0 &&  task->prio <= 99  /
{
  $delta = nsecs - @start[tid];
  printf("nanosleep() - %d nsecs\n", $delta);
  delete(@start[tid]);
}
```

Calculate the delta

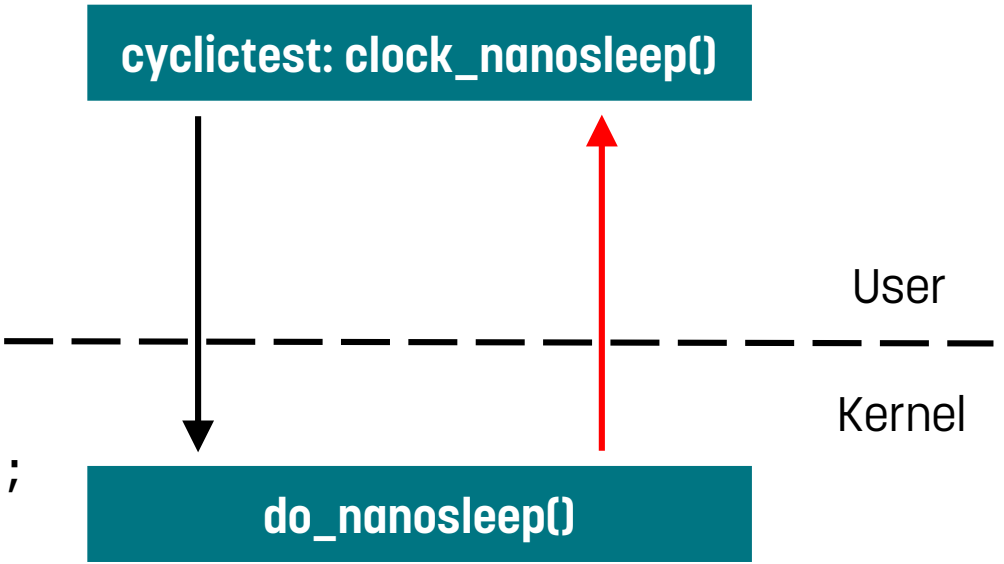


bpftrace: clock_nanosleep()

```
kprobe:do_nanosleep
/ comm == "cyclictest" && curtask->prio <= 99 /
{
  @start[tid] = nsecs;
}

kretpr
/ comm == "cyc                  start[tid]
    != 0 &&      task->prio <= 99  /
{
  $delta = nsecs - @start[tid];
  printf("nanosleep() - %d nsecs\n", $delta);
  delete(@start[tid]);
}
```

Print the delta to stdout

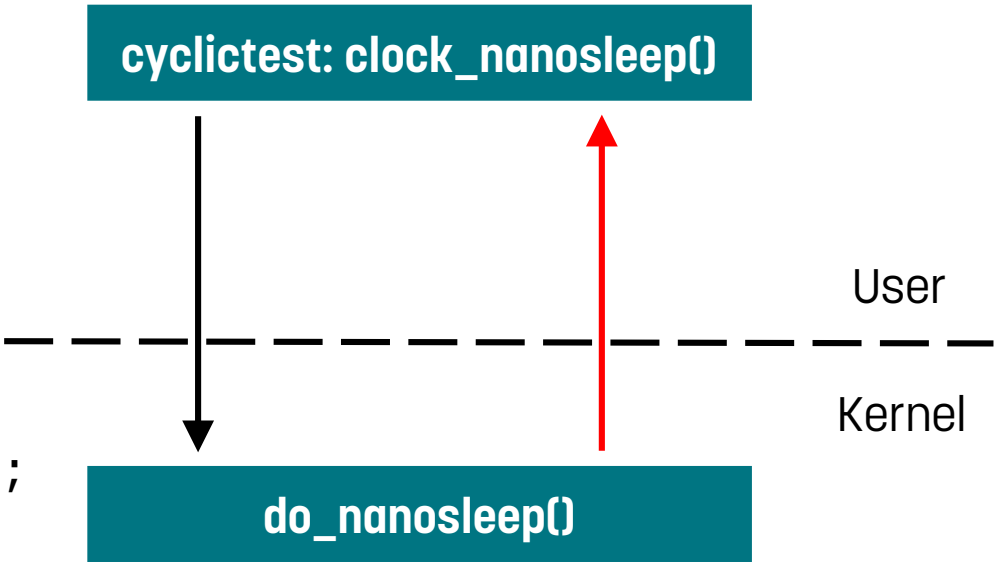


bpftrace: clock_nanosleep()

```
kprobe:do_nanosleep
/ comm == "cyclictest" && curtask->prio <= 99 /
{
  @start[tid] = nsecs;
}

kretpr
/ comm == "cyc
    != 0 && c    k->prio <= 99  /
{
  $delta =  secs - @start[tid];
  printf( nanosleep() - %d nsecs\n", $delta);
  delete(@start[tid]);
}
```

Remove the timestamp from the list



bpftrace: Identify the root cause

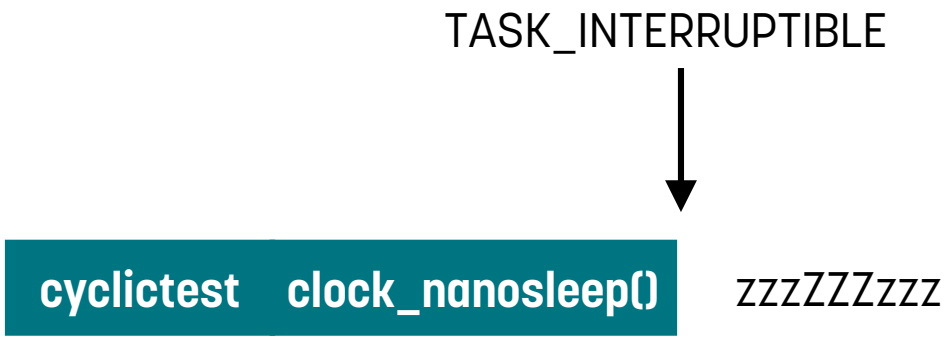
```
kprobe:do_nanosleep
/ comm == "cyclictest" && curtask->prio <= 99 /
{
    @start_sleep[tid] = nsecs;
}
[...]
```

LET'S ADD SOME ADDITIONAL LOGIC!

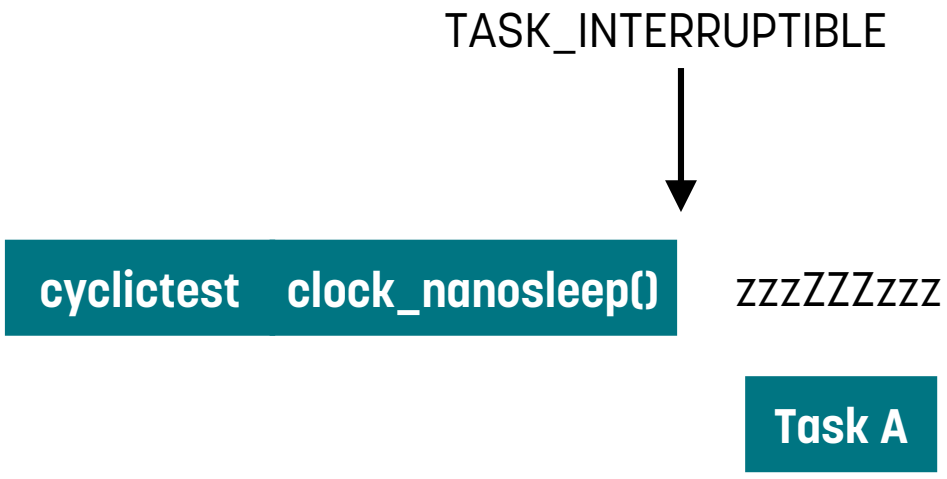
Scheduling events

cyclictest **clock_nanosleep()**

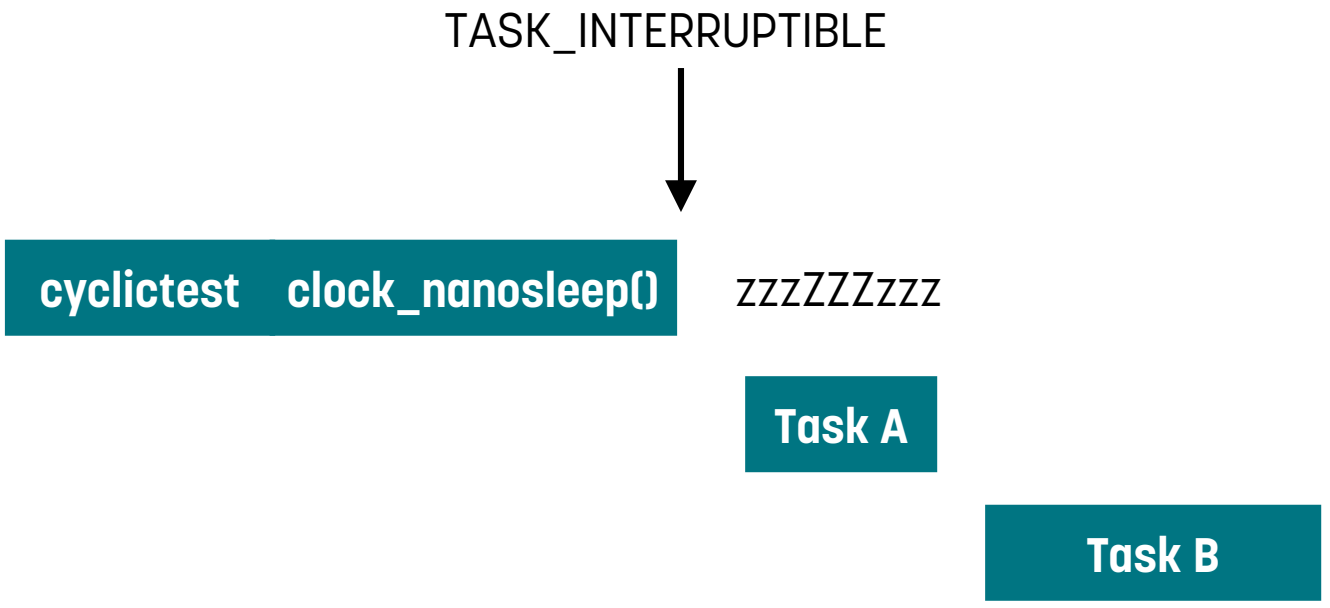
Scheduling events



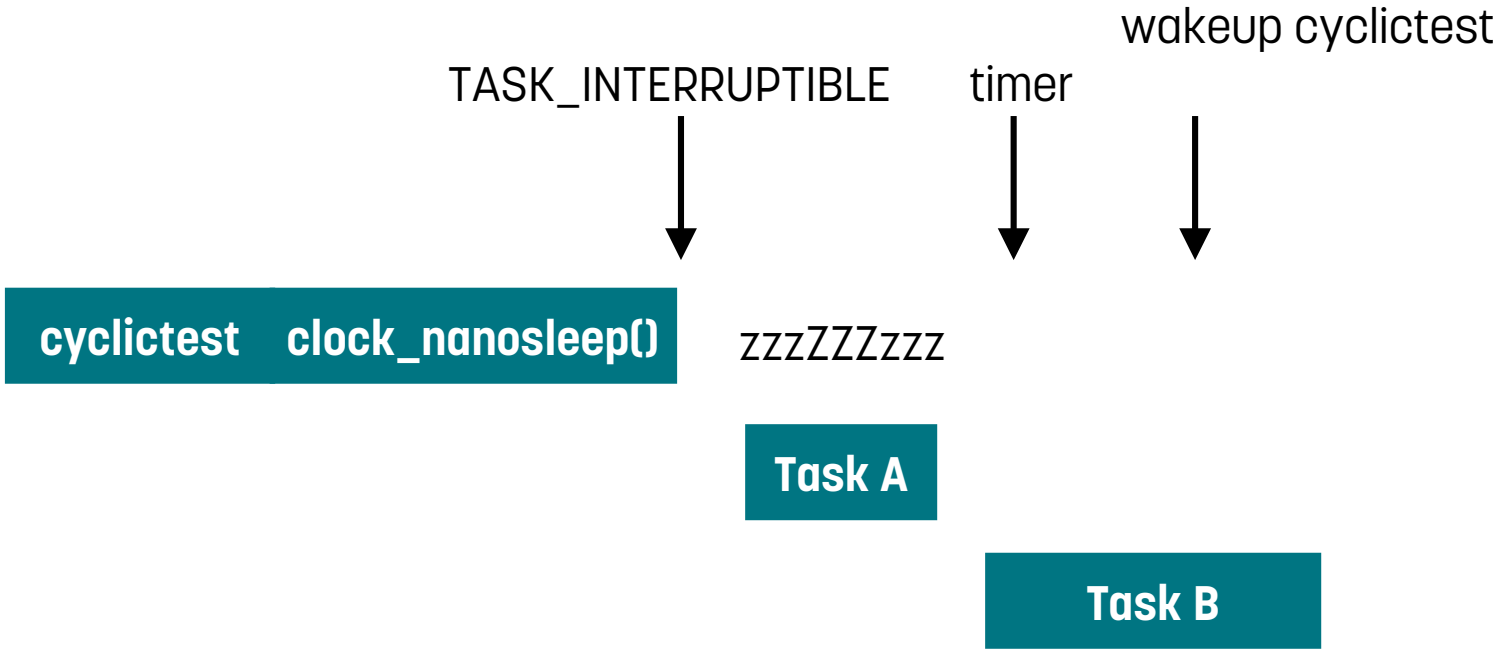
Scheduling events



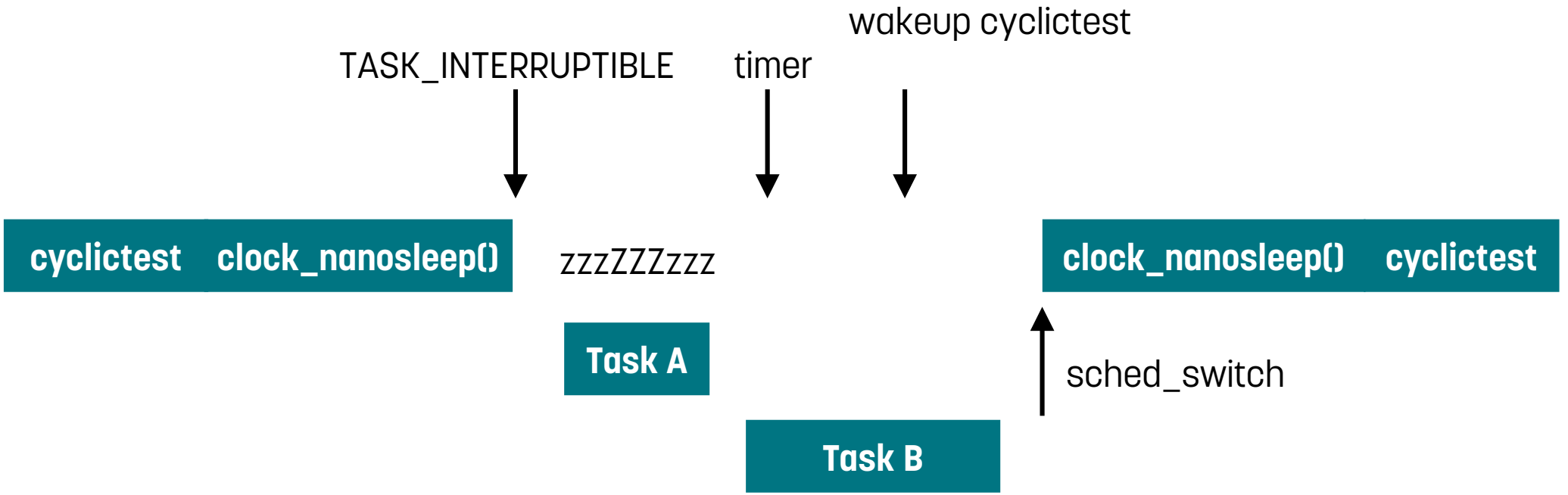
Scheduling events



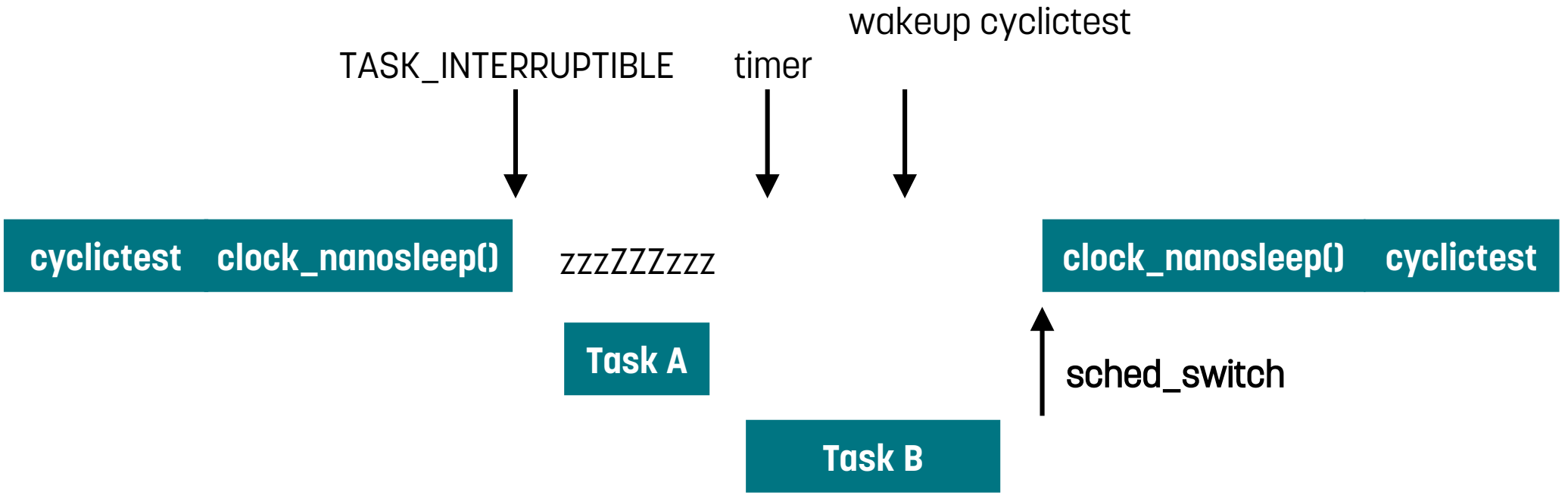
Scheduling events



Scheduling events



Scheduling events



bpftrace: Identify the root cause

```
tracepoint:sched:sched_switch
/ args->next_comm == "cyclictest" && args->next_prio <= 99 &&
            @start[args->next_pid] != 0 /
{
   @prevcomm[args->next_pid] = args->prev_comm;
   @prevpid[args->next_pid] = args->prev_pid;
   @prevprio[args->next_pid] = args->prev_prio;
   @switchtime[args->next_pid] = nsecs;
}
```

bpftrace: Identify the root cause

```
tracepoint:sched:sched_switch
/ args->next_comm == "cyclictest" && args->next_prio <= 99 &&
            @start[args->next_pid] != 0 /
{
   @prevcomm[args->next_pid] = args
   @prevpid[args->next_pid] = arg
   @prevprio[args->next_pid] = args
   @switchtime[args->next_pid] = nsecs;
}
```

Let's use the sched_switch event

bpftrace: Identify the root cause

```
tracepoint:sched:sched_switch
/ args->next_comm == "cyclictest" && args->next_prio <= 99 &&
            @start[args->next_pid] != 0 /
{
   @prevcomm[args->next_pid] = args
   @prevpid[args->next_pid] = arg
   @prevprio[args->next_pid] = args
   @switchtime[args->next_pid] = nsecs;
}
```

If cyclictest is scheduled next

bpftrace: Identify the root cause

```
tracepoint:sched:sched_switch
/ args->next_comm == "cyclictest" && args->next_prio <= 99 &&
            @start[args->next_pid] != 0 /
{
   @prevcomm[args->next_pid] = args
   @prevpid[args->next_pid] = arg
   @prevprio[args->next_pid] = args
   @switchtime[args->next_pid] = nsecs;
}
```

and the thread has RT priority

bpftrace: Identify the root cause

```
tracepoint:sched:sched_switch
/ args->next_comm == "cyclictest" && args->next_prio <= 99 &&
            @start[args->next_pid] != 0 /
{
   @prevcomm[args->next_pid] = args
   @prevpid[args->next_pid] = arg
   @prevprio[args->next_pid] = args
   @switchtime[args->next_pid] = nsecs;
}
```

and we have a nanosleep() active for this thread

bpftrace: Identify the root cause

```
tracepoint:sched:sched_switch
/ args->next_comm == "cyclictest" && args->next
            @start[args->next_pid] != 0 /
{
   @prevcomm[args->next_pid] = args->prev_comm;
   @prevpid[args->next_pid] = args->prev_pid;
   @prevprio[args->next_pid] = args->prev_prio;
   @switchtime[args->next_pid] = nsecs;
}
```

We fetch name, PID and priority of the task which was active before

bpftrace: Identify the root cause

```
kretprobe:do_nanosleep
/ comm == "cyclictest" && @start[tid] != 0 && curtask->prio <= 99  /
{
    $delta = (nsecs - @start[tid]) / 1000;
    $latency = $delta - $1;

    if ($delta > $1 && $latency > $2) {
        printf("nanosleep() - %d usecs\n", $delta);
        printf("->  prev_comm %s <- possible culprit\n", @prevcomm[tid]);
        printf("->  prev_pid %d\n", @prevpid[tid]);
        printf("->  prev_prio %d \n", @prevprio[tid]);
    }

[...]
}
```

bpftrace: Identify the root cause

```
kretprobe:do_nanosleep
/ comm == "cyclictest" && @start[tid] != 0
{
    $delta = (nsecs - @start[tid]) / 1
    $latency = $delta - $1;

    if ($delta > $1 && $latency > $2) {
        printf("nanosleep() - %d usecs\n", $delta);
        printf("->  prev_comm %s <- possible culprit\n", @prevcomm[tid]);
        printf("->  prev_pid %d\n", @prevpid[tid]);
        printf("->  prev_prio %d \n", @prevprio[tid]);
    }

[...]
}
```

Timer interval (via commandline)

bpftrace: Identify the root cause

```
kretprobe:do_nanosleep
/ comm == "cyclictest" && @start[tid] != 0
{
    $delta = (nsecs - @start[tid]) / 1
    $latency = $delta - $1;

    if ($delta > $1 && $latency > $2) {
        printf("nanosleep() - %d usecs\n", $delta);
        printf("->  prev_comm %s <- possible culprit\n", @prevcomm[tid]);
        printf("->  prev_pid %d\n", @prevpid[tid]);
        printf("->  prev_prio %d \n", @prevprio[tid]);
    }
[...]
}
```

Latency threshold (via commandline)

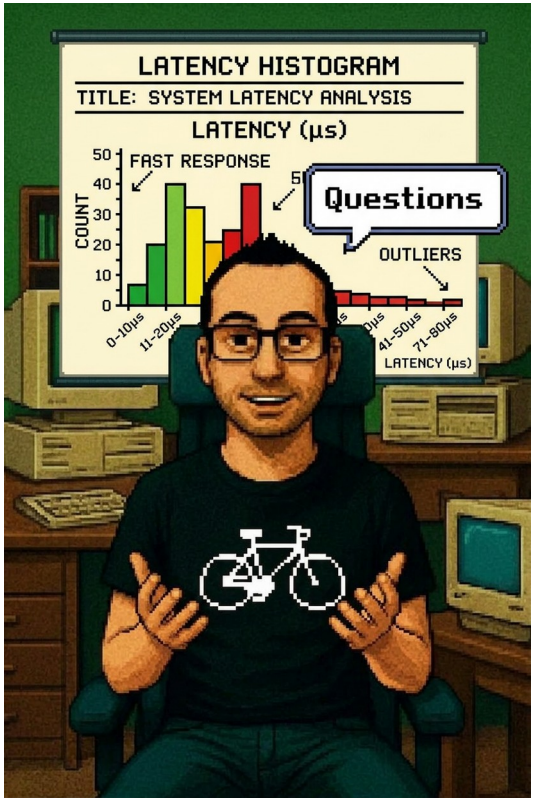
bpftrace: Identify the root cause

```
# ./nanosleep.bt 200 1000
Attaching 3 probes...

[...]

nanosleep() - 4854033 nsecs
->  prev_comm mklatency
->  prev_pid 8848
->  prev_prio 120
[...]
```

Questions?



AI generated content (Canva Business)