Tracing on Embedded Boards

For limited systems

Why trace on embedded?

- Tracing is great for debugging new development

Why trace on embedded?

- Tracing is great for debugging new development
- Embedded boards have a lot of corner cases

Why trace on embedded?

- Tracing is great for debugging new development
- Embedded boards have a lot of corner cases
- Visibility into the happenings of the board is key for solving issues

Introducing `trace-cmd`

Introducing `trace-cmd`

- Command line tool for ftrace

Introducing `trace-cmd`

- Command line tool for ftrace
    - You know what ftrace is right?

Introducing `trace-cmd`

- Command line tool for ftrace
  - You know what ftrace is right?
  - If not, then watch previous talks from https://kernel-recipes.org

Introducing `trace-cmd`

- Command line tool for ftrace
  - You know what ftrace is right?
  - If not, then watch previous talks from https://kernel-recipes.org
- Starts / stops tracing

Introducing `trace-cmd`

- Command line tool for ftrace
  - You know what ftrace is right?
  - If not, then watch previous talks from https://kernel-recipes.org
- Starts / stops tracing
- Shows the current trace buffer

Introducing `trace-cmd`

- Command line tool for ftrace
  - You know what ftrace is right?
  - If not, then watch previous talks from https://kernel-recipes.org
- Starts / stops tracing
- Shows the current trace buffer
- Records into a data file for post analysis (and to pass it around)

Introducing `trace-cmd`

- Command line tool for ftrace
    - You know what ftrace is right?
    - If not, then watch previous talks from https://kernel-recipes.org
- Starts / stops tracing
- Shows the current trace buffer
- Records into a data file for post analysis (and to pass it around)
- Reads the data file

Introducing `trace-cmd`

- Command line tool for ftrace
  - You know what ftrace is right?
  - If not, then watch previous talks from https://kernel-recipes.org
- Starts / stops tracing
- Shows the current trace buffer
- Records into a data file for post analysis (and to pass it around)
- Reads the data file
- Much much more
- Full man pages at:
  - https://www.trace-cmd.org/Documentation/trace-cmd/

Example tracing

```
# trace-cmd start -p function -l ‘*lock*’ -n ‘*clock*’
# trace-cmd show
# tracer: function
#
# entries-in-buffer/entries-written: 66426/87886   #P:8
#
#                                _-----=> irqs-off/BH-disabled
#                               / _----=> need-resched
#                              | / _---=> hardirq/softirq
#                              || / _--=> preempt-depth
#                              ||| / _-=> migrate-disable
#                              |||| /     delay
#           TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#              | |         |   |||||     |         |
          <idle>-0       [006] d..3.  7857.354704: rcu_read_lock_sched_held <-lock_release
          <idle>-0       [006] d..3.  7857.354704: rcu_read_lock_sched_held <-lock_acquire
          <idle>-0       [006] d..3.  7857.354704: rcu_read_lock_sched_held <-lock_release
          <idle>-0       [006] d..2.  7857.354705: _raw_spin_lock <-get_next_timer_interrupt
          <idle>-0       [006] d..4.  7857.354705: rcu_read_lock_sched_held <-lock_acquire
          <idle>-0       [006] d..3.  7857.354705: do_raw_spin_trylock <-_raw_spin_lock
```

Example tracing (without filters)

```
# trace-cmd start -p function
# trace-cmd show
# tracer: function
#
# entries-in-buffer/entries-written: 370446/38974811   #P:8
[..]
#           TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#              | |         |   |||||     |         |
          <idle>-0       [003] ..s3.  1526.601458: rcu_read_lock_sched_held <-__do_softirq
          <idle>-0       [003] d.s2.  1526.601459: irqtime_account_irq <-__do_softirq
          <idle>-0       [003] d.s2.  1526.601459: __local_bh_enable <-__do_softirq
          <idle>-0       [003] d..2.  1526.601460: idle_cpu <-__irq_exit_rcu
          <idle>-0       [003] d..2.  1526.601460: tick_nohz_irq_exit <-irq_exit_rcu
          <idle>-0       [003] d..2.  1526.601460: ktime_get <-tick_nohz_irq_exit
          <idle>-0       [003] d..3.  1526.601460: rcu_read_lock_sched_held <-lock_acquire
          <idle>-0       [003] d..3.  1526.601460: rcu_read_lock_sched_held <-lock_release
          <idle>-0       [003] d..3.  1526.601461: rcu_read_lock_sched_held <-trace_hardirqs_on_prep
          <idle>-0       [003] ...2.  1526.601462: cpuidle_reflect <-cpuidle_idle_call
          <idle>-0       [003] ...2.  1526.601462: menu_reflect <-cpuidle_idle_call
          <idle>-0       [003] ...2.  1526.601462: tick_nohz_idle_got_tick <-menu_reflect
          <idle>-0       [003] ...2.  1526.601462: arch_cpu_idle_exit <-do_idle
          <idle>-0       [003] d..2.  1526.601463: arch_cpu_idle_enter <-do_idle
          <idle>-0       [003] d..2.  1526.601463: tsc_verify_tsc_adjust <-arch_cpu_idle_enter
```

Example tracing (function graph)

```
# trace-cmd start -p function_graph
# trace-cmd show
# tracer: function_graph
#
# CPU  DURATION                  FUNCTION CALLS
# |     |   |                     |   |   |   |
 3)   3.480 us    |  rcu_idle_exit();
 3)   0.449 us    |  rcu_read_lock_sched_held();
 3)   0.283 us    |  sched_idle_set_state();
 3)               |  cpuidle_reflect() {
 3)               |    menu_reflect() {
 3)   0.321 us    |      tick_nohz_idle_got_tick();
 3)   1.057 us    |    }
 3)   2.068 us    |  }
 3)   0.301 us    |  arch_cpu_idle_exit();
 3)               |  tick_nohz_idle_exit() {
 3)               |    ktime_get() {
 3)   0.306 us    |      rcu_read_lock_sched_held();
 3)   0.271 us    |      rcu_read_lock_sched_held();
 3)   1.957 us    |    }
 3)   0.261 us    |    nr_iowait_cpu();
 3)               |    __tick_nohz_full_update_tick() {
 3)   0.316 us    |      check_tick_dependency();
```

trace-cmd allows for offline analysis

- Can record to a file `trace.dat`

trace-cmd allows for offline analysis

- Can record to a file `trace.dat`
- Can add plugins to process events (do things differently)

trace-cmd allows for offline analysis

- Can record to a file `trace.dat`
- Can add plugins to process events (do things differently)
- Can use tooling to analyse the data
    - The data file is basically a database of events

Recording trace (function graph)

```
# trace-cmd set -n '*[^c]lock*'
# trace-cmd record -p function_graph
[Ctrl^C]
# trace-cmd report
      trace-cmd-2612  [000]  7514.222672: funcgraph_entry:                   |    update_rq_clock() {
      trace-cmd-2612  [000]  7514.222673: funcgraph_exit:         0.236 us   |    }
      trace-cmd-2612  [000]  7514.222673: funcgraph_entry:                   |    dequeue_task_fair() {
      trace-cmd-2612  [000]  7514.222673: funcgraph_entry:                   |      dequeue_entity() {
      trace-cmd-2612  [000]  7514.222673: funcgraph_entry:                   |        update_curr() {
      trace-cmd-2612  [000]  7514.222674: funcgraph_entry:                   |          update_min_vruntime() {
      trace-cmd-2612  [000]  7514.222674: funcgraph_exit:         0.210 us   |          }
      trace-cmd-2612  [000]  7514.222674: funcgraph_entry:                   |          cpuacct_charge() {
      trace-cmd-2612  [000]  7514.222674: funcgraph_exit:         0.211 us   |          }
      trace-cmd-2612  [000]  7514.222674: funcgraph_entry:                   |          __cgroup_account_cputime() {
      trace-cmd-2612  [000]  7514.222675: funcgraph_entry:                   |            preempt_count_add() {
      trace-cmd-2612  [000]  7514.222675: funcgraph_exit:         0.193 us   |            }
      trace-cmd-2612  [000]  7514.222675: funcgraph_entry:                   |            cgroup_rstat_updated() {
      trace-cmd-2612  [000]  7514.222675: funcgraph_exit:         0.190 us   |            }
      trace-cmd-2612  [000]  7514.222675: funcgraph_entry:                   |            preempt_count_sub() {
      trace-cmd-2612  [000]  7514.222676: funcgraph_exit:         0.198 us   |            }
      trace-cmd-2612  [000]  7514.222676: funcgraph_exit:         1.309 us   |          }
      trace-cmd-2612  [000]  7514.222676: funcgraph_exit:         2.554 us   |        }
      trace-cmd-2612  [000]  7514.222676: funcgraph_entry:                   |        __update_load_avg_se() {
      trace-cmd-2612  [000]  7514.222676: funcgraph_exit:         0.197 us   |        }
      trace-cmd-2612  [000]  7514.222676: funcgraph_entry:                   |        __update_load_avg_cfs_rq() {
      trace-cmd-2612  [000]  7514.222677: funcgraph_exit:         0.214 us   |        }
      trace-cmd-2612  [000]  7514.222678: funcgraph_entry:                   |        clear_buddies() {
```

Recording trace (function)

```
# trace-cmd record -p function -n '*[^c]lock*'
[Ctrl^C]
# trace-cmd report
       trace-cmd-2640  [000]  7621.673360: function:             handle_mm_fault
       trace-cmd-2640  [000]  7621.673360: function:                mem_cgroup_from_task
       trace-cmd-2640  [000]  7621.673360: function:             __count_memcg_events
       trace-cmd-2640  [000]  7621.673360: function:                cgroup_rstat_updated
       trace-cmd-2640  [000]  7621.673361: function:             __handle_mm_fault
       trace-cmd-2640  [000]  7621.673361: function:                handle_pte_fault
       trace-cmd-2640  [000]  7621.673361: function:                do_fault
       trace-cmd-2640  [000]  7621.673361: function:                   do_read_fault
       trace-cmd-2640  [000]  7621.673361: function:                      filemap_map_pages
       trace-cmd-2640  [000]  7621.673361: function:                         next_uptodate_page
       trace-cmd-2640  [000]  7621.673361: function:                         filemap_map_pmd
       trace-cmd-2640  [000]  7621.673370: function:                         PageHeadHuge
       trace-cmd-2640  [000]  7621.673370: function:                         do_set_pte
       trace-cmd-2640  [000]  7621.673371: function:                            add_mm_counter_fast
       trace-cmd-2640  [000]  7621.673371: function:                            page_add_file_rmap
       trace-cmd-2640  [000]  7621.673371: function:                               lock_page_memcg
       trace-cmd-2640  [000]  7621.673371: function:                         next_uptodate_page
       trace-cmd-2640  [000]  7621.673371: function:                         PageHeadHuge
       trace-cmd-2640  [000]  7621.673371: function:                         do_set_pte
       trace-cmd-2640  [000]  7621.673371: function:                            add_mm_counter_fast
       trace-cmd-2640  [000]  7621.673371: function:                            page_add_file_rmap
       trace-cmd-2640  [000]  7621.673371: function:                               lock_page_memcg
       trace-cmd-2640  [000]  7621.673371: function:                         next_uptodate_page
       trace-cmd-2640  [000]  7621.673372: function:                         PageHeadHuge
```

Example tracing (without filters)

```
# trace-cmd start -p function
# trace-cmd show
# tracer: function
#
# entries-in-buffer/entries-written: 370446/38974811   #P:8
[..]
#          TASK-PID     CPU#  |||||  TIMESTAMP  FUNCTION
#             | |         |   |||||     |         |
          <idle>-0      [003] ..s3.  1526.601458: rcu_read_lock_sched_held <-__do_softirq
          <idle>-0      [003] d.s2.  1526.601459: irqtime_account_irq <-__do_softirq
          <idle>-0      [003] d.s2.  1526.601459: __local_bh_enable <-__do_softirq
          <idle>-0      [003] d..2.  1526.601460: idle_cpu <-__irq_exit_rcu
          <idle>-0      [003] d..2.  1526.601460: tick_nohz_irq_exit <-irq_exit_rcu
          <idle>-0      [003] d..2.  1526.601460: ktime_get <-tick_nohz_irq_exit
          <idle>-0      [003] d..3.  1526.601460: rcu_read_lock_sched_held <-lock_acquire
          <idle>-0      [003] d..3.  1526.601460: rcu_read_lock_sched_held <-lock_release
          <idle>-0      [003] d..3.  1526.601461: rcu_read_lock_sched_held <-trace_hardirqs_on_prep
          <idle>-0      [003] ...2.  1526.601462: cpuidle_reflect <-cpuidle_idle_call
          <idle>-0      [003] ...2.  1526.601462: menu_reflect <-cpuidle_idle_call
          <idle>-0      [003] ...2.  1526.601462: tick_nohz_idle_got_tick <-menu_reflect
          <idle>-0      [003] ...2.  1526.601462: arch_cpu_idle_exit <-do_idle
          <idle>-0      [003] d..2.  1526.601463: arch_cpu_idle_enter <-do_idle
          <idle>-0      [003] d..2.  1526.601463: tsc_verify_tsc_adjust <-arch_cpu_idle_enter
```

Recording trace (function)

```
# trace-cmd record -p function -n '*[^c]lock*'
[Ctrl^C]
# trace-cmd report
       trace-cmd-2640  [000]  7621.673360: function:             handle_mm_fault
       trace-cmd-2640  [000]  7621.673360: function:                mem_cgroup_from_task
       trace-cmd-2640  [000]  7621.673360: function:             __count_memcg_events
       trace-cmd-2640  [000]  7621.673360: function:                cgroup_rstat_updated
       trace-cmd-2640  [000]  7621.673361: function:             __handle_mm_fault
       trace-cmd-2640  [000]  7621.673361: function:                handle_pte_fault
       trace-cmd-2640  [000]  7621.673361: function:                do_fault
       trace-cmd-2640  [000]  7621.673361: function:                   do_read_fault
       trace-cmd-2640  [000]  7621.673361: function:                      filemap_map_pages
       trace-cmd-2640  [000]  7621.673361: function:                         next_uptodate_page
       trace-cmd-2640  [000]  7621.673361: function:                         filemap_map_pmd
       trace-cmd-2640  [000]  7621.673370: function:                         PageHeadHuge
       trace-cmd-2640  [000]  7621.673370: function:                         do_set_pte
       trace-cmd-2640  [000]  7621.673371: function:                            add_mm_counter_fast
       trace-cmd-2640  [000]  7621.673371: function:                            page_add_file_rmap
       trace-cmd-2640  [000]  7621.673371: function:                               lock_page_memcg
       trace-cmd-2640  [000]  7621.673371: function:                         next_uptodate_page
       trace-cmd-2640  [000]  7621.673371: function:                         PageHeadHuge
       trace-cmd-2640  [000]  7621.673371: function:                         do_set_pte
       trace-cmd-2640  [000]  7621.673371: function:                            add_mm_counter_fast
       trace-cmd-2640  [000]  7621.673371: function:                            page_add_file_rmap
       trace-cmd-2640  [000]  7621.673371: function:                               lock_page_memcg
       trace-cmd-2640  [000]  7621.673371: function:                         next_uptodate_page
       trace-cmd-2640  [000]  7621.673372: function:                         PageHeadHuge
```

Recording trace (function)

```
# trace-cmd record -p function -n '*[^c]lock*'
[Ctrl^C]
# trace-cmd report -O parent
      trace-cmd-2640  [000]  7621.673360: function:             handle_mm_fault <-- do_user_addr_fault
      trace-cmd-2640  [000]  7621.673360: function:                mem_cgroup_from_task <-- handle_mm_fault
      trace-cmd-2640  [000]  7621.673360: function:             __count_memcg_events <-- count_memcg_events.constprop.0
      trace-cmd-2640  [000]  7621.673360: function:                cgroup_rstat_updated <-- __count_memcg_events
      trace-cmd-2640  [000]  7621.673361: function:             __handle_mm_fault <-- handle_mm_fault
      trace-cmd-2640  [000]  7621.673361: function:                handle_pte_fault <-- __handle_mm_fault
      trace-cmd-2640  [000]  7621.673361: function:                do_fault <-- __handle_mm_fault
      trace-cmd-2640  [000]  7621.673361: function:                   do_read_fault <-- do_fault
      trace-cmd-2640  [000]  7621.673361: function:                      filemap_map_pages <-- do_read_fault
      trace-cmd-2640  [000]  7621.673361: function:                         next_uptodate_page <-- filemap_map_pages
      trace-cmd-2640  [000]  7621.673361: function:                         filemap_map_pmd <-- filemap_map_pages
      trace-cmd-2640  [000]  7621.673370: function:                         PageHeadHuge <-- filemap_map_pages
      trace-cmd-2640  [000]  7621.673370: function:                         do_set_pte <-- filemap_map_pages
      trace-cmd-2640  [000]  7621.673371: function:                            add_mm_counter_fast <-- do_set_pte
      trace-cmd-2640  [000]  7621.673371: function:                            page_add_file_rmap <-- do_set_pte
      trace-cmd-2640  [000]  7621.673371: function:                               lock_page_memcg <-- page_add_file_rmap
      trace-cmd-2640  [000]  7621.673371: function:                         next_uptodate_page <-- filemap_map_pages
      trace-cmd-2640  [000]  7621.673371: function:                         PageHeadHuge <-- filemap_map_pages
      trace-cmd-2640  [000]  7621.673371: function:                         do_set_pte <-- filemap_map_pages
      trace-cmd-2640  [000]  7621.673371: function:                            add_mm_counter_fast <-- do_set_pte
      trace-cmd-2640  [000]  7621.673371: function:                            page_add_file_rmap <-- do_set_pte
      trace-cmd-2640  [000]  7621.673371: function:                               lock_page_memcg <-- page_add_file_rmap
      trace-cmd-2640  [000]  7621.673371: function:                         next_uptodate_page <-- filemap_map_pages
      trace-cmd-2640  [000]  7621.673372: function:                         PageHeadHuge <-- filemap_map_pages
```

Recording trace (function)

```
# trace-cmd record -p function -n '*[^c]lock*'
[Ctrl^C]
# trace-cmd report -O parent
       trace-cmd-2640  [000]  7621.673360: function:             handle_mm_fault <-- do_user_addr_fault
       trace-cmd-2640  [000]  7621.673360: function:                mem_cgroup_from_task <-- handle_mm_fault
       trace-cmd-2640  [000]  7621.673360: function:             __count_memcg_events <-- count_memcg_events.constprop.0
       trace-cmd-2640  [000]  7621.673360: function:                cgroup_rstat_updated <-- __count_memcg_events
       trace-cmd-2640  [000]  7621.673361: function:             __handle_mm_fault <-- handle_mm_fault
       trace-cmd-2640  [000]  7621.673361: function:                handle_pte_fault <-- __handle_mm_fault
       trace-cmd-2640  [000]  7621.673361: function:                do_fault <-- __handle_mm_fault
       trace-cmd-2640  [000]  7621.673361: function:                   do_read_fault <-- do_fault
       trace-cmd-2640  [000]  7621.673361: function:                      filemap_map_pages <-- do_read_fault
       trace-cmd-2640  [000]  7621.673361: function:                         next_uptodate_page <-- filemap_map_pages
       trace-cmd-2640  [000]  7621.673361: function:                         filemap_map_pmd <-- filemap_map_pages
       trace-cmd-2640  [000]  7621.673370: function:                         PageHeadHuge <-- filemap_map_pages
       trace-cmd-2640  [000]  7621.673370: function:                         do_set_pte <-- filemap_map_pages
       trace-cmd-2640  [000]  7621.673371: function:                            add_mm_counter_fast <-- do_set_pte
       trace-cmd-2640  [000]  7621.673371: function:                            page_add_file_rmap <-- do_set_pte
       trace-cmd-2640  [000]  7621.673371: function:                               lock_page_memcg <-- page_add_file_rmap
       trace-cmd-2640  [000]  7621.673371: function:                         next_uptodate_page <-- filemap_map_pages
       trace-cmd-2640  [000]  7621.673371: function:                         PageHeadHuge <-- filemap_map_pages
       trace-cmd-2640  [000]  7621.673371: function:                         do_set_pte <-- filemap_map_pages
       trace-cmd-2640  [000]  7621.673371: function:                            add_mm_counter_fast <-- do_set_pte
       trace-cmd-2640  [000]  7621.673371: function:                            page_add_file_rmap <-- do_set_pte
       trace-cmd-2640  [000]  7621.673371: function:                               lock_page_memcg <-- page_add_file_rmap
       trace-cmd-2640  [000]  7621.673371: function:                         next_uptodate_page <-- filemap_map_pages
       trace-cmd-2640  [000]  7621.673372: function:                         PageHeadHuge <-- filemap_map_pages
```

trace-cmd trace.dat files

- Really useful to have

trace-cmd trace.dat files

- Really useful to have
  - Offline analysis

trace-cmd trace.dat files

- Really useful to have
    - Offline analysis
    - Run queries on events

trace-cmd trace.dat files

- Really useful to have
  - Offline analysis
  - Run queries on events
    - Make histograms

trace-cmd trace.dat files

- Really useful to have
  - Offline analysis
  - Run queries on events
    - Make histograms
    - Show latency between related events

trace-cmd trace.dat files

- Really useful to have
  - Offline analysis
  - Run queries on events
    - Make histograms
    - Show latency between related events
- Records much more than what the kernel buffer can hold

trace-cmd trace.dat files

- Really useful to have
  - Offline analysis
  - Run queries on events
    - Make histograms
    - Show latency between related events
- Records much more than what the kernel buffer can hold
  - Trace long sessions

trace-cmd trace.dat files

- Really useful to have
  - Offline analysis
  - Run queries on events
    - Make histograms
    - Show latency between related events
- Records much more than what the kernel buffer can hold
  - Trace long sessions
  - Uses splice(2) system call (more on this later)

trace-cmd trace.dat files

- Really useful to have
  - Offline analysis
  - Run queries on events
    - Make histograms
    - Show latency between related events
- Records much more than what the kernel buffer can hold
  - Trace long sessions
  - Uses splice(2) system call (more on this later)
    - Makes it fast!

splice(2) system call

splice(2) system call

- Connects a file descriptor with a pipe

splice(2) system call

- Connects a file descriptor with a pipe
- Allows transfer of data without copying to/from user space

splice(2) system call

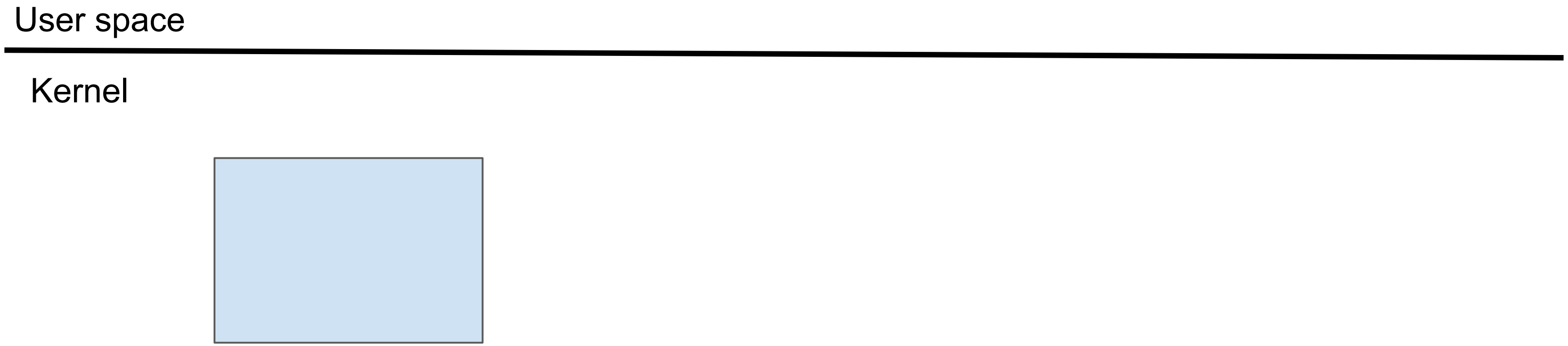
- Connects a file descriptor with a pipe
- Allows transfer of data without copying to/from user space
- Moves pages around inside the kernel without copying

Simple “cp” program

```
int main(int argc, char **argv)
{
	char buf[BUFSIZ];
	int ifd, ofd, r;

	if (argc < 3)
		exit(-1);
	ifd = open(argv[1], O_RDONLY);
	if (ifd < 0)
		exit(-1);
	ofd = open(argv[2], O_WRONLY | O_TRUNC | O_CREAT, 0644);
	if (ofd < 0)
		exit(-1);

	while ((r = read(ifd, buf, BUFSIZ)) > 0) {
		r = write(ofd, buf, r);
		if (r < 0)
			exit(-1);
	}

	close(ifd);
	close(ofd);
	exit(0);
}
```

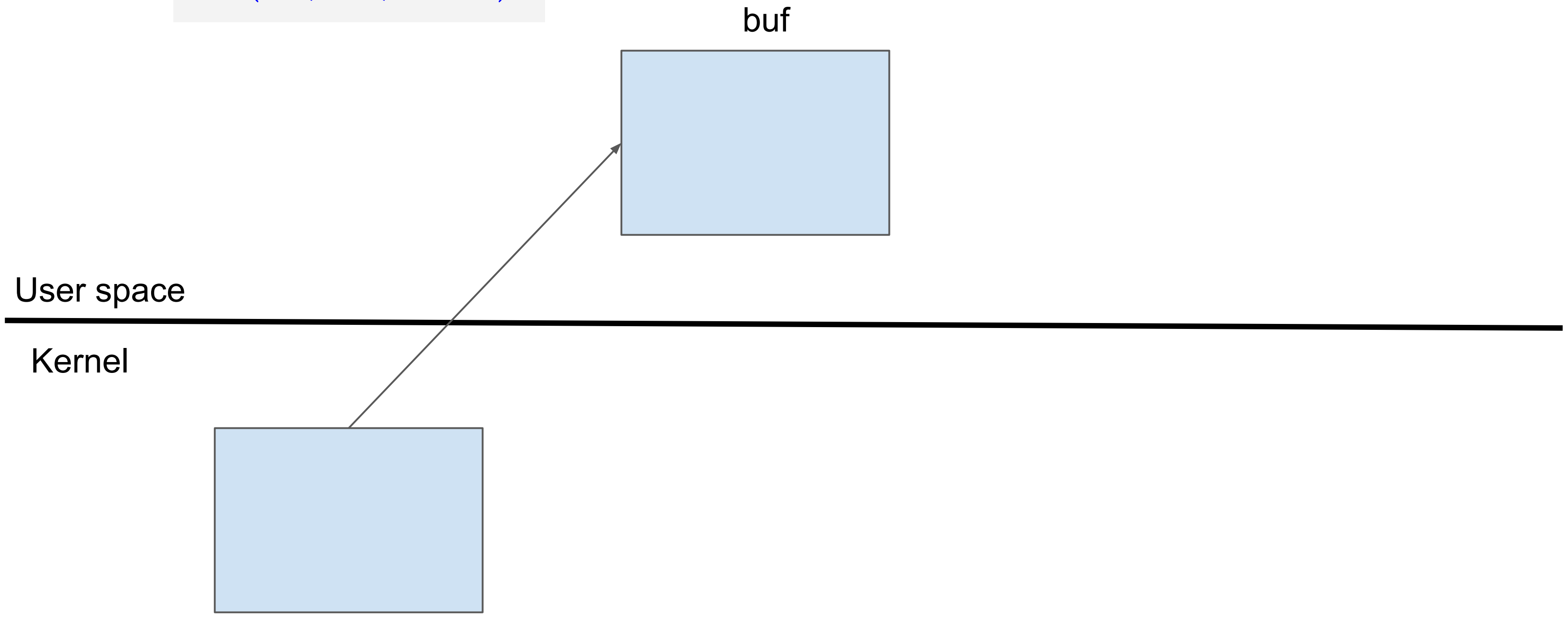
Simple “cp” program

```
read(ifd, buf, BUFSIZ)
```



Simple “cp” program

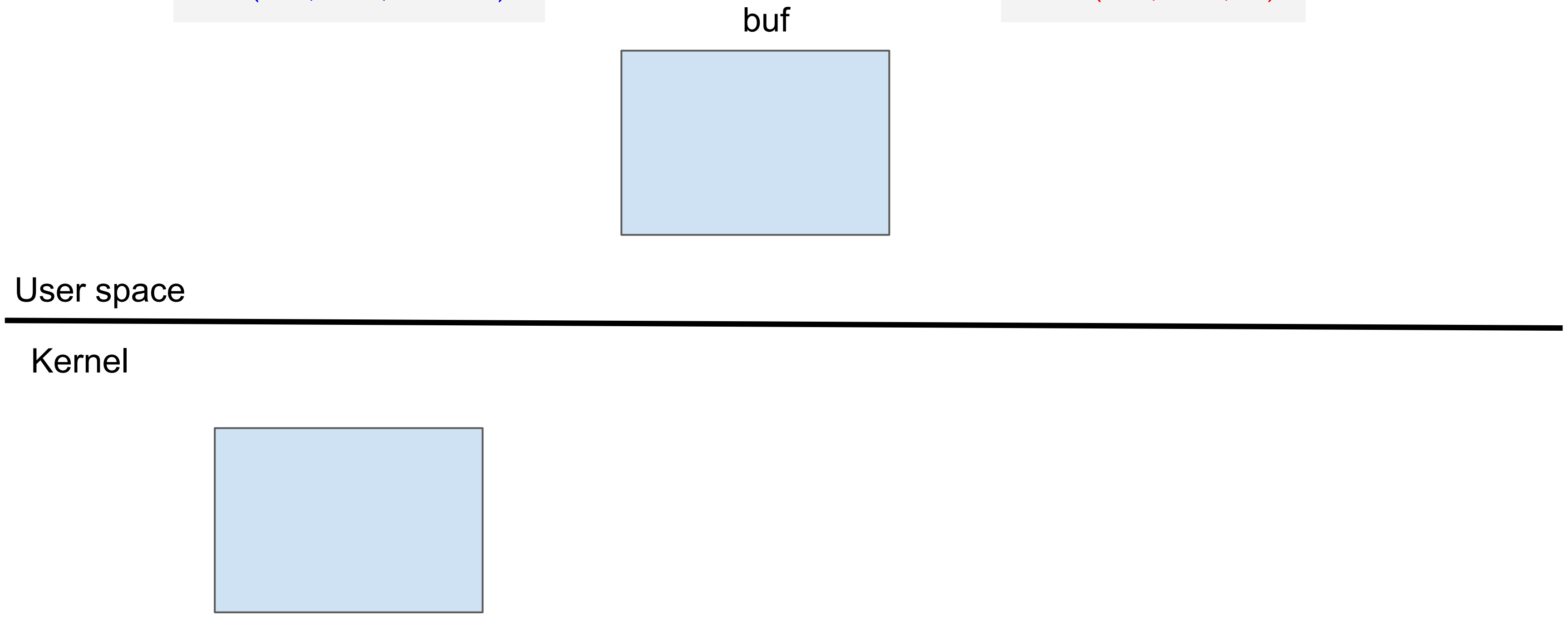
```
read(ifd, buf, BUFSIZ)
```



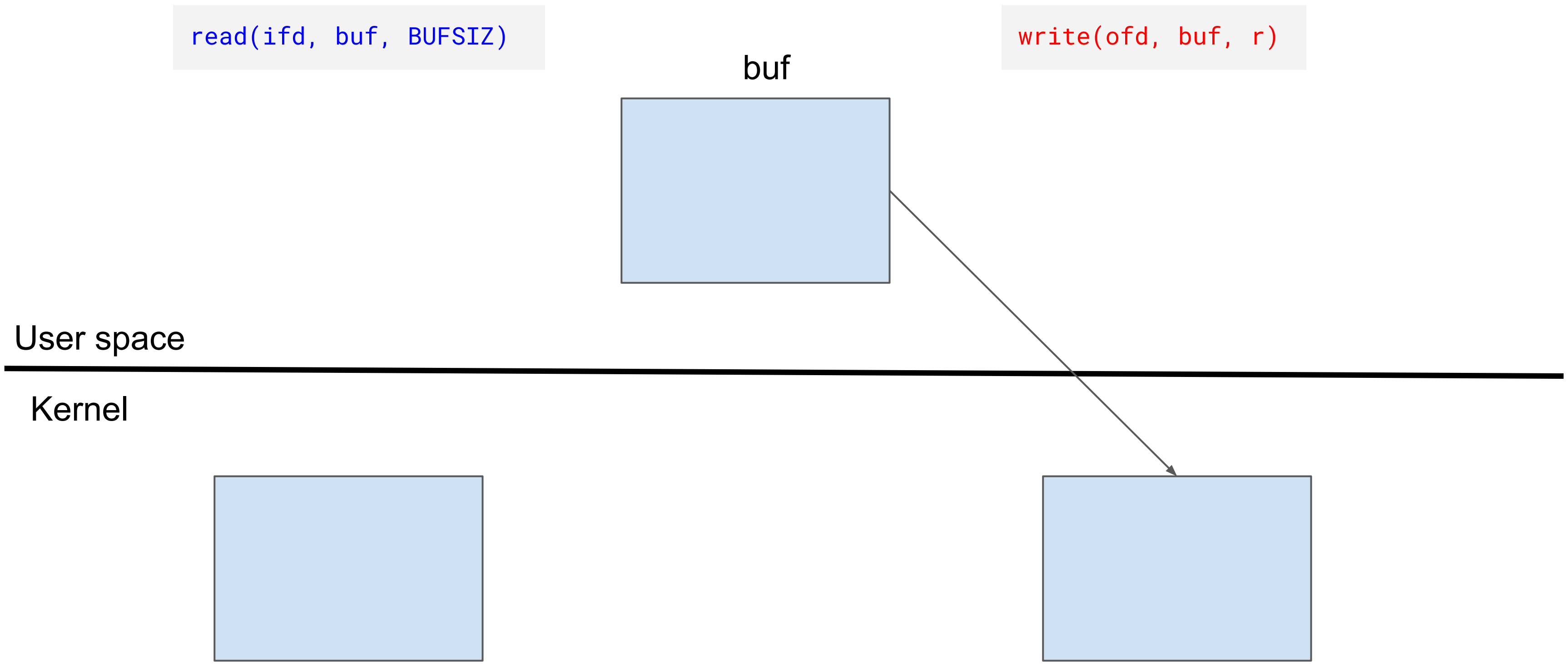
Simple “cp” program

```
read(ifd, buf, BUFSIZ)
```

```
write(ofd, buf, r)
```



Simple “cp” program

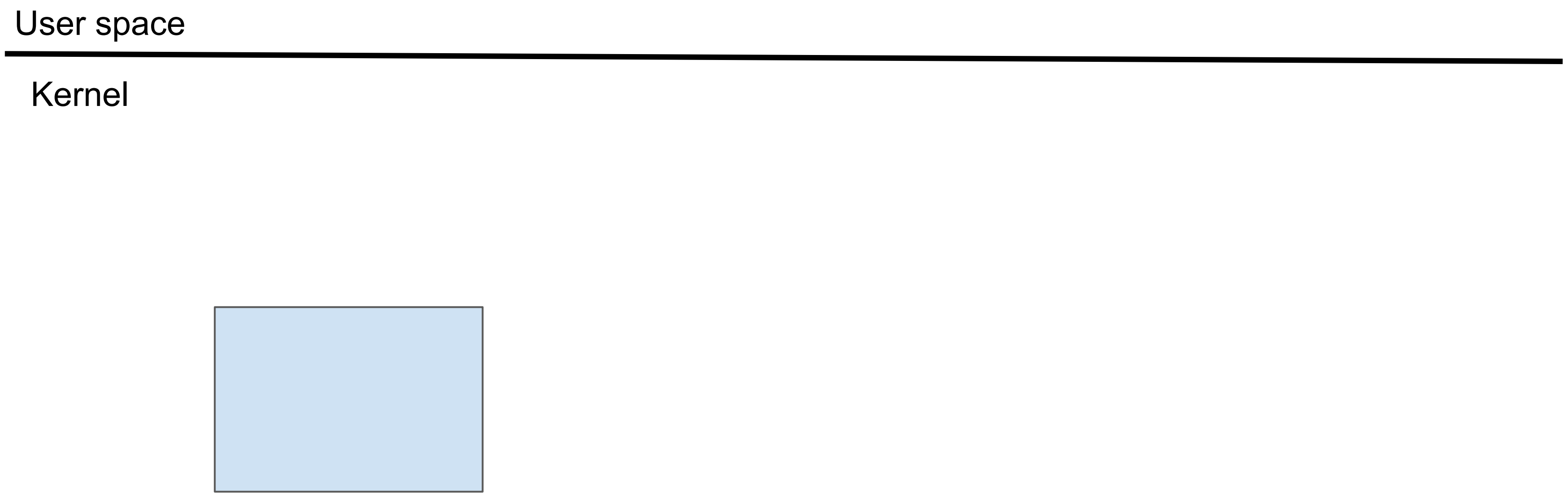


“cp” program with splice

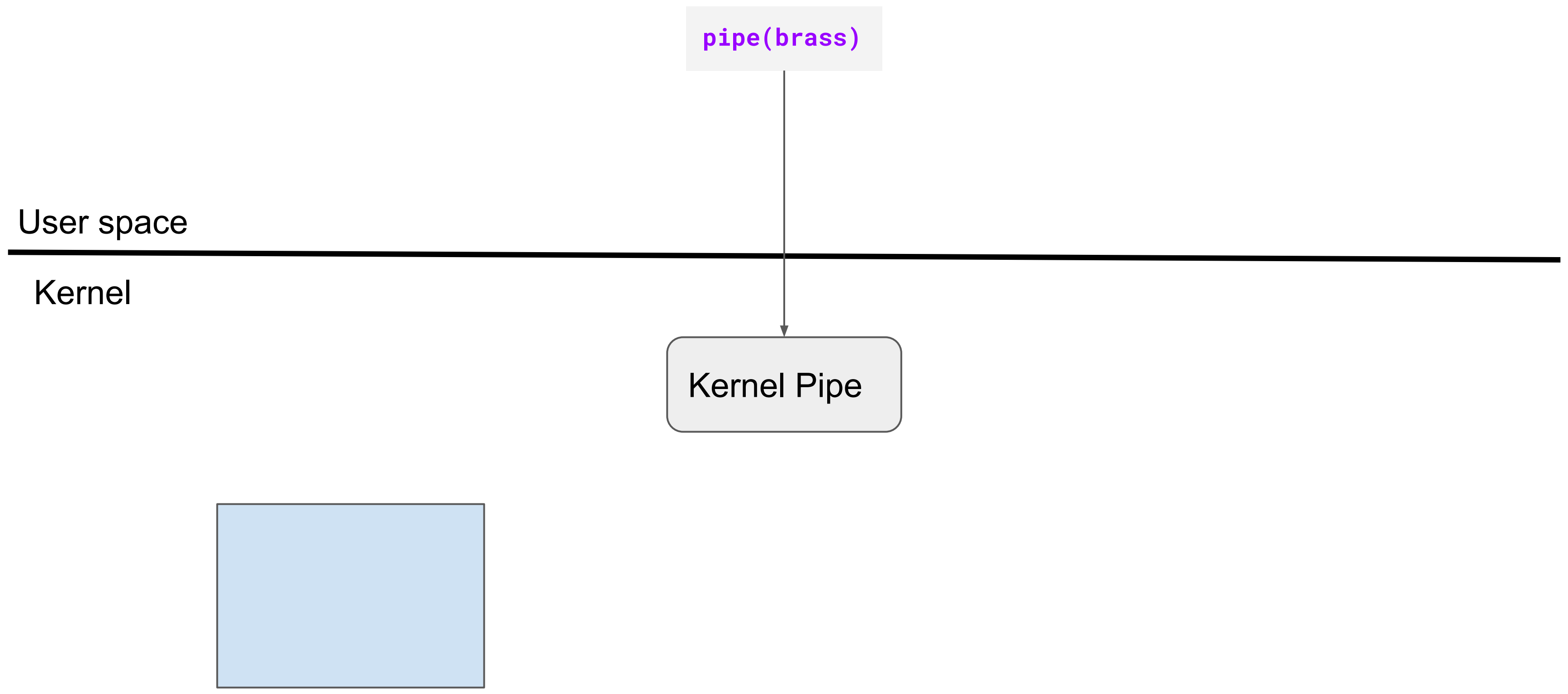
```
int main(int argc, char **argv)
{
        int ifd, ofd, r;
        int brass[2];
        int pipesize;

        if (argc < 3)
                exit(-1);
        ifd = open(argv[1], O_RDONLY);
        if (ifd < 0)
                exit(-1);
        ofd = open(argv[2], O_WRONLY | O_TRUNC | O_CREAT, 0644);
        if (ofd < 0)
                exit(-1);
        if (pipe(brass) < 0)
                exit(-1);
        r = fcntl(brass[0], F_GETPIPE_SZ, &pipesize);
        if (r < 0)
                pipesize = getpagesize();
        for (;;) {
                r = splice(ifd, NULL, brass[1], NULL, pipesize, SPLICE_F_MOVE);
                if (r < 0)
                        exit(-1);
                if (!r)
                        break;
                r = splice(brass[0], NULL, ofd, NULL, r, SPLICE_F_MOVE | SPLICE_F_NONBLOCK);
                if (r < 0)
                        exit(-1);
        }
        close(ifd);
        close(ofd);
        exit(0);
}
```

“cp” program with splice

```
pipe(brass)
```



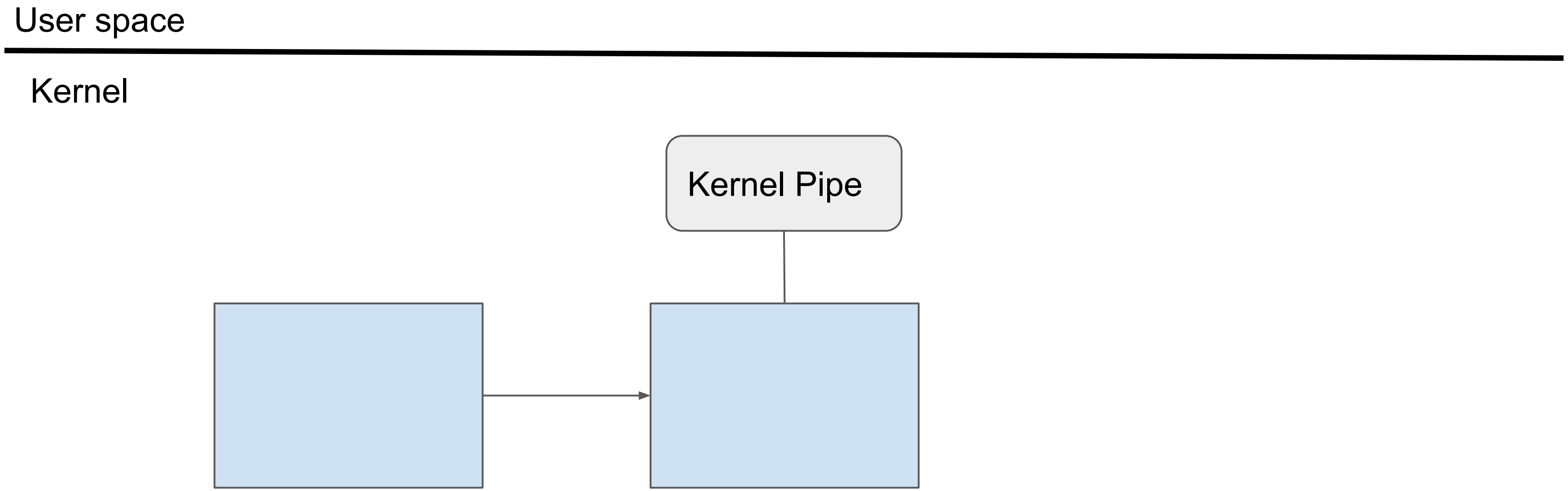
“cp” program with splice



“cp” program with splice

```
pipe(brass)
```

```
splice(ifd,.., brass[1])
```



“cp” program with splice

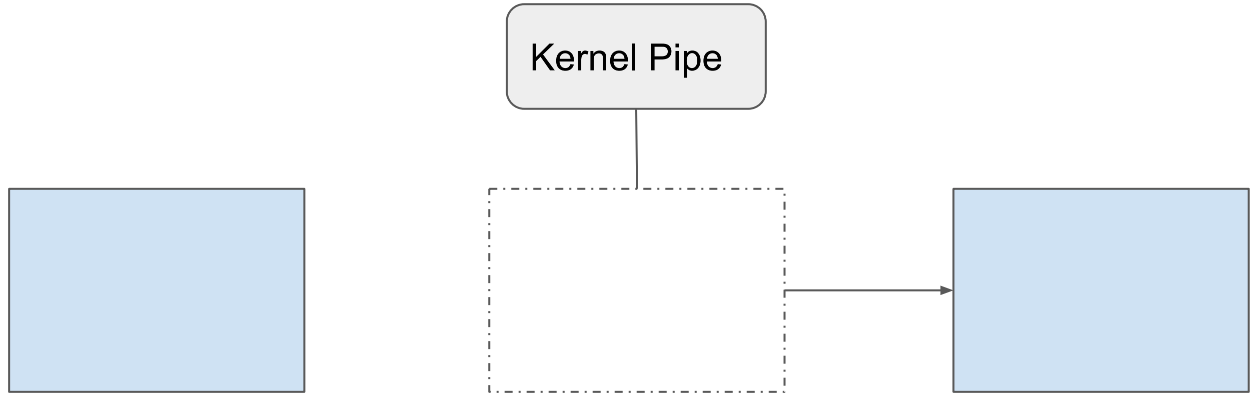
```
pipe(brass)
```

```
splice(ifd,.., brass[1])
```

```
splice(brass[0], .., ofp)
```

User space

Kernel

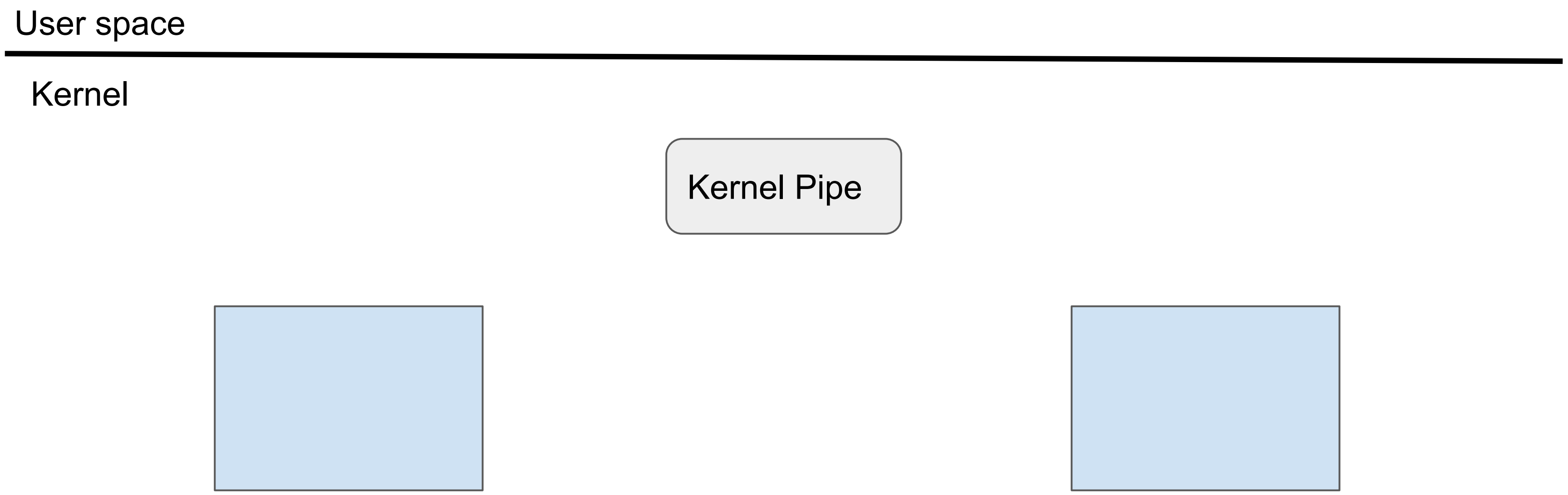


“cp” program with splice

```
pipe(brass)
```

```
splice(ifd,.., brass[1])
```

```
splice(brass[0], .., ofp)
```



Ftrace and splice

Ftrace and splice

- The ftrace ring buffers were developed with splice in mind

Ftrace and splice

- The ftrace ring buffers were developed with splice in mind
- The ring buffer is split into sub buffers (currently architecture page size)

Ftrace and splice

- The ftrace ring buffers were developed with splice in mind
- The ring buffer is split into sub buffers (currently architecture page size)
- The sub buffers can be swapped out with new pages
    - Writers never block
    - Readers do a cmpxchg spin to swap out a data page

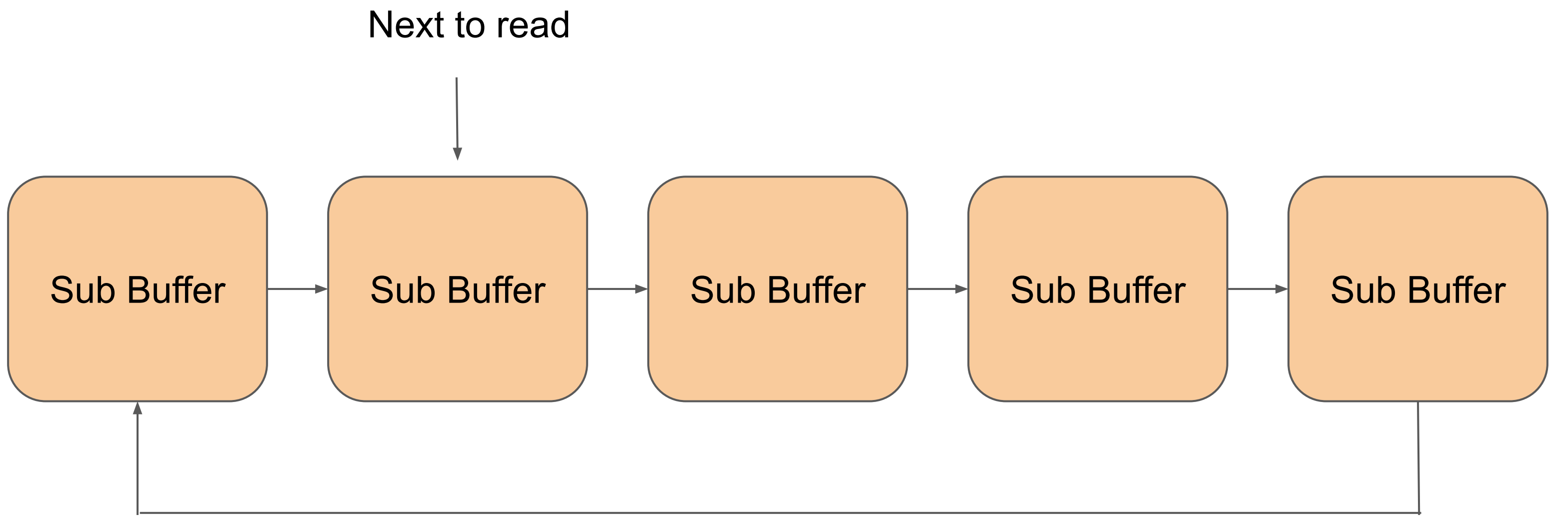
Ftrace and splice

- The ftrace ring buffers were developed with splice in mind
- The ring buffer is split into sub buffers (currently architecture page size)
- The sub buffers can be swapped out with new pages
  - Writers never block
  - Readers do a cmpxchg spin to swap out a data page
- The trace data is never copied (Written by event, passed via splice to the file)

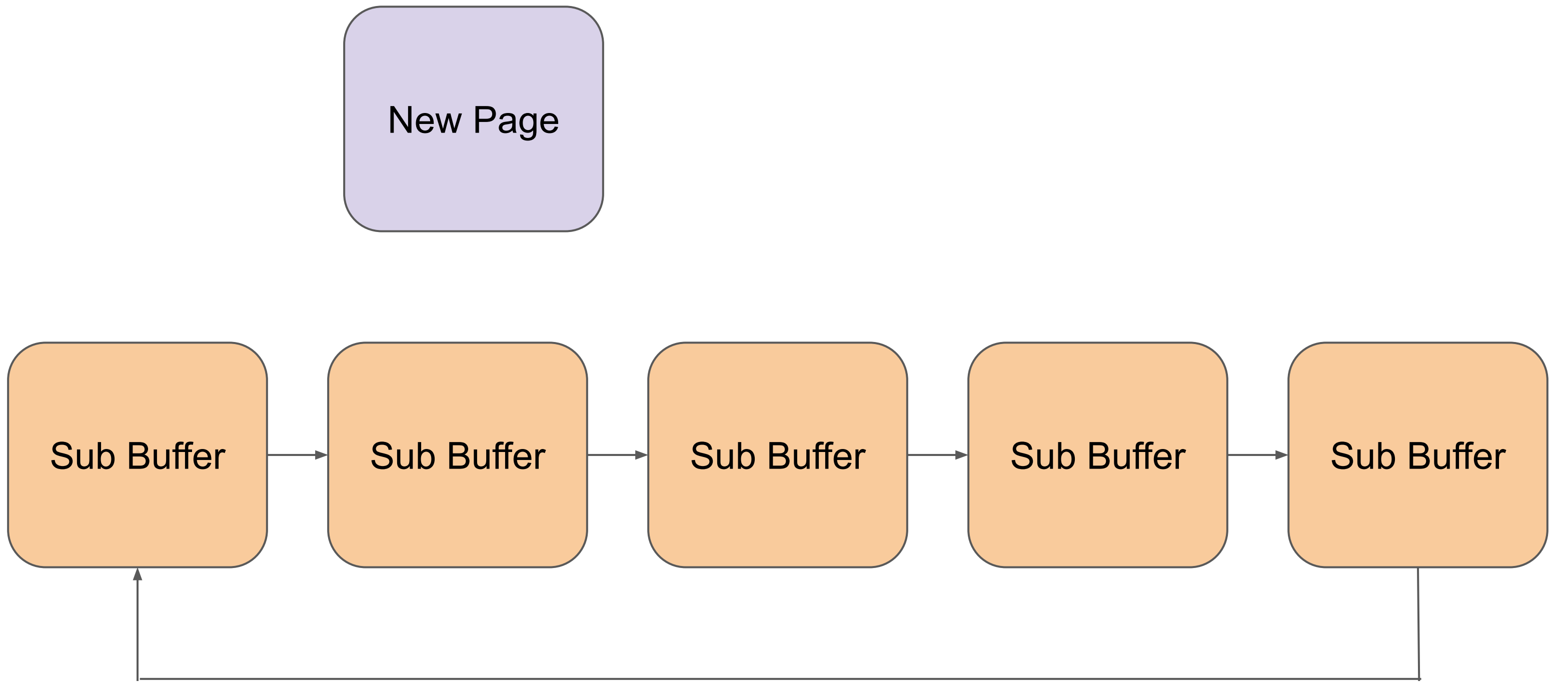
Ftrace and splice

- The ftrace ring buffers were developed with splice in mind
- The ring buffer is split into sub buffers (currently architecture page size)
- The sub buffers can be swapped out with new pages
  - Writers never block
  - Readers do a cmpxchg spin to swap out a data page
- The trace data is never copied (Written by event, passed via splice to the file)
  - True zero copy!

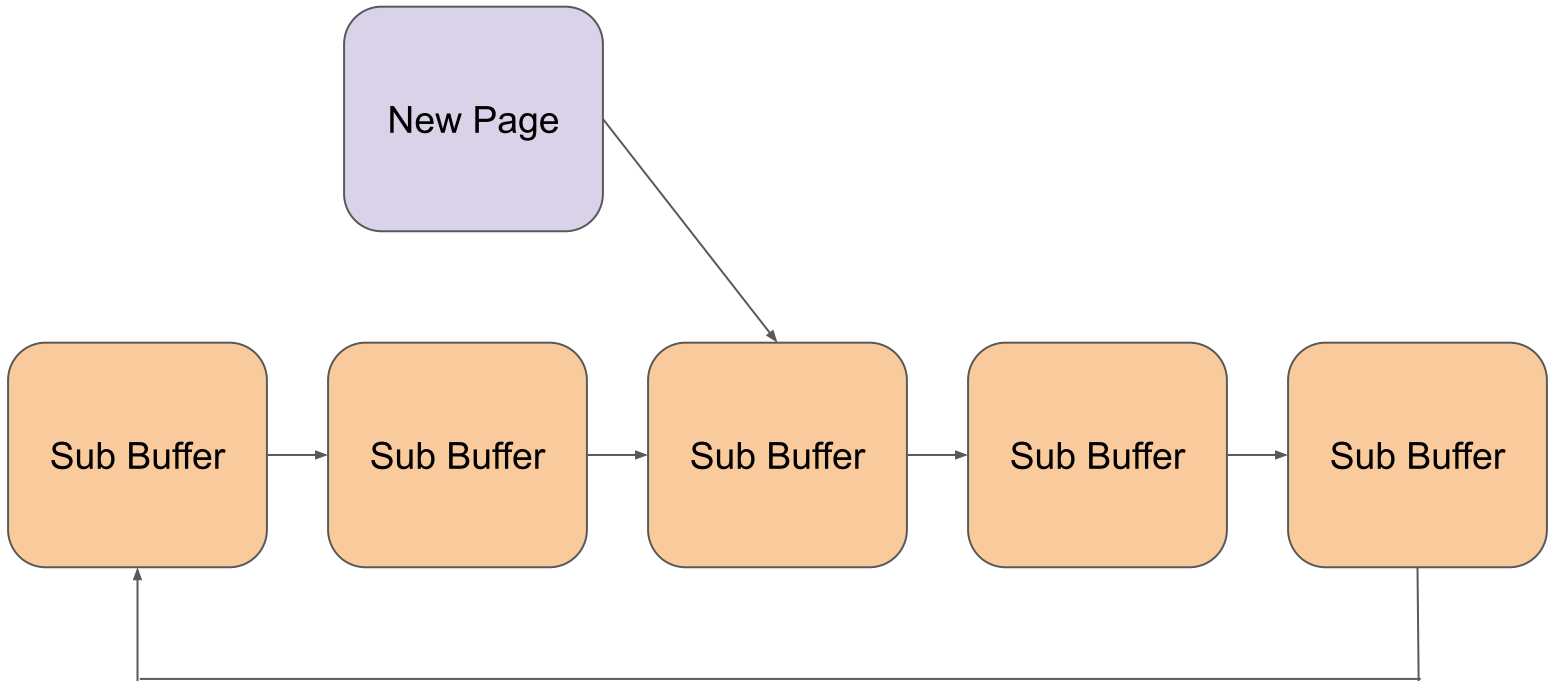
Ftrace ring buffer and splice



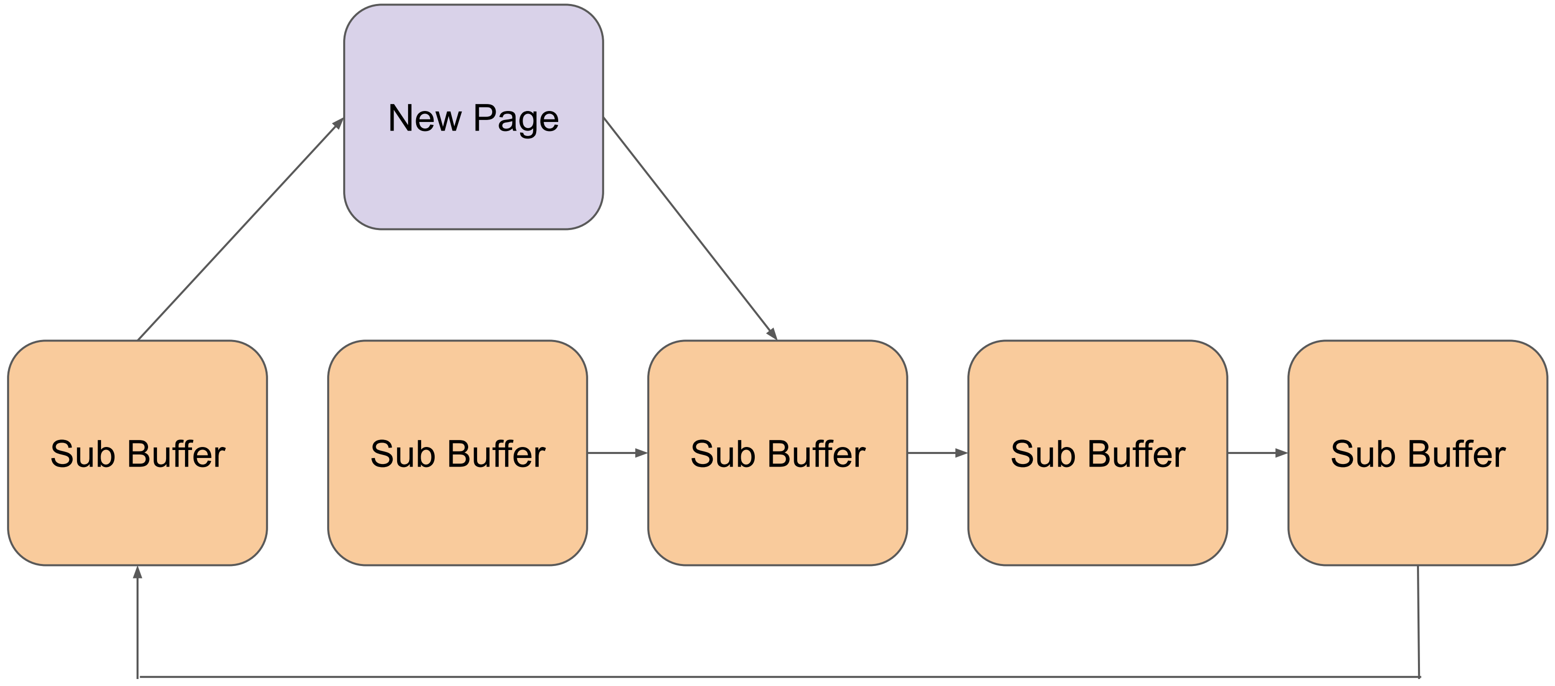
Ftrace ring buffer and splice



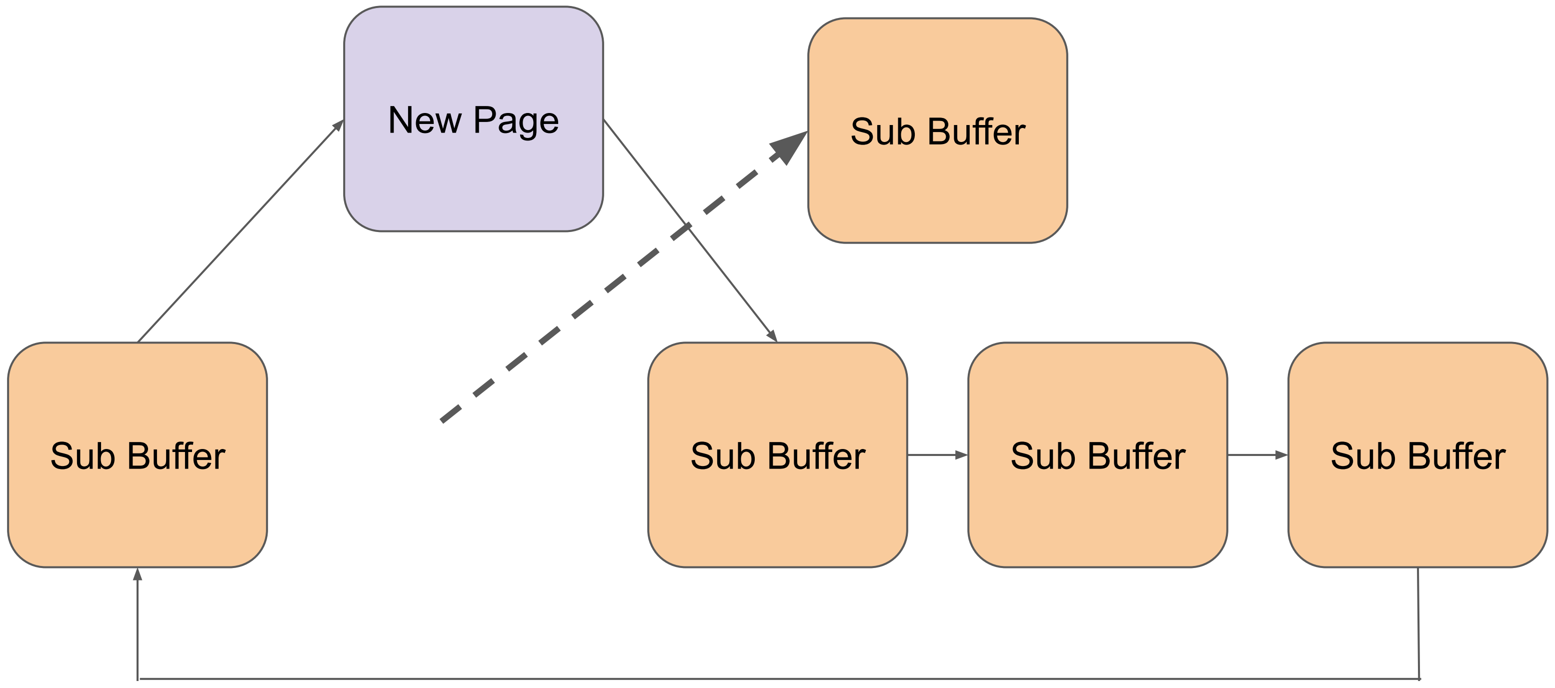
Ftrace ring buffer and splice



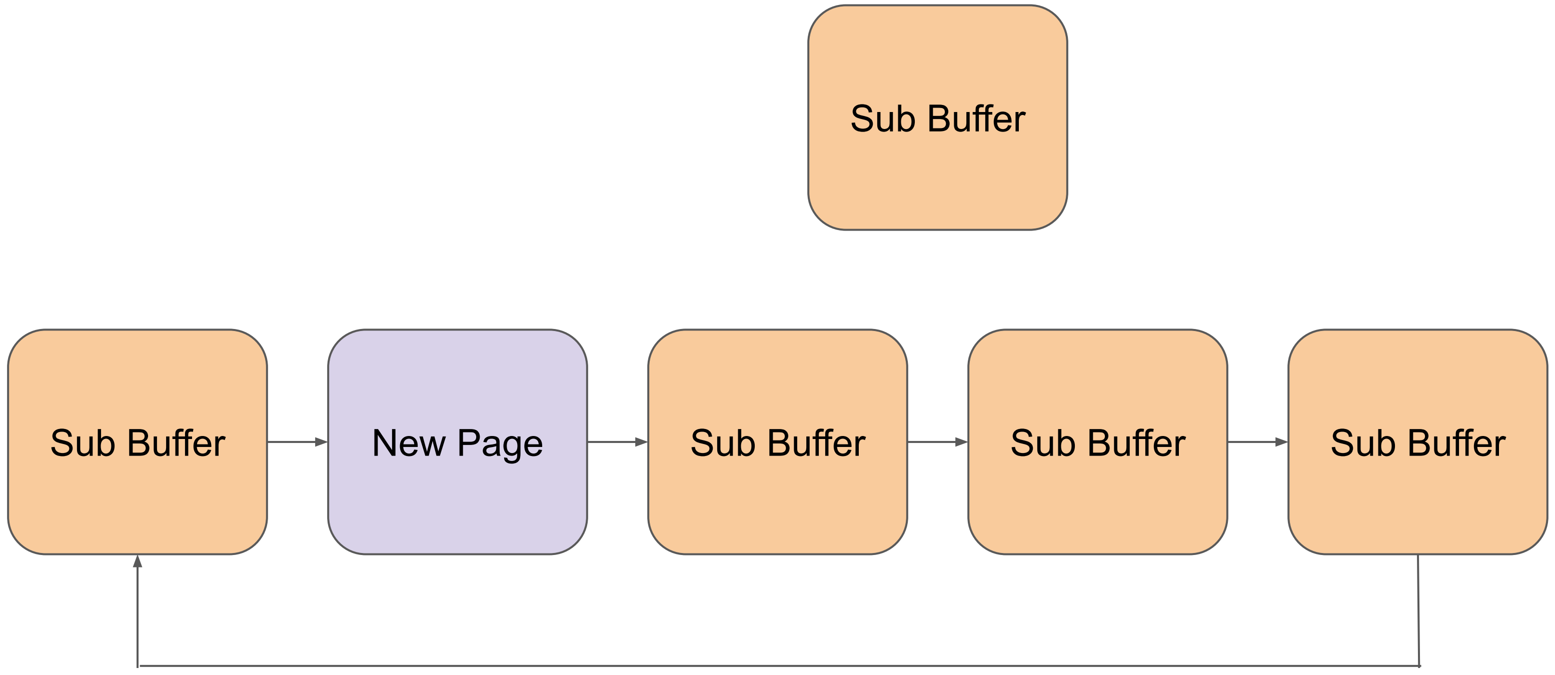
Ftrace ring buffer and splice



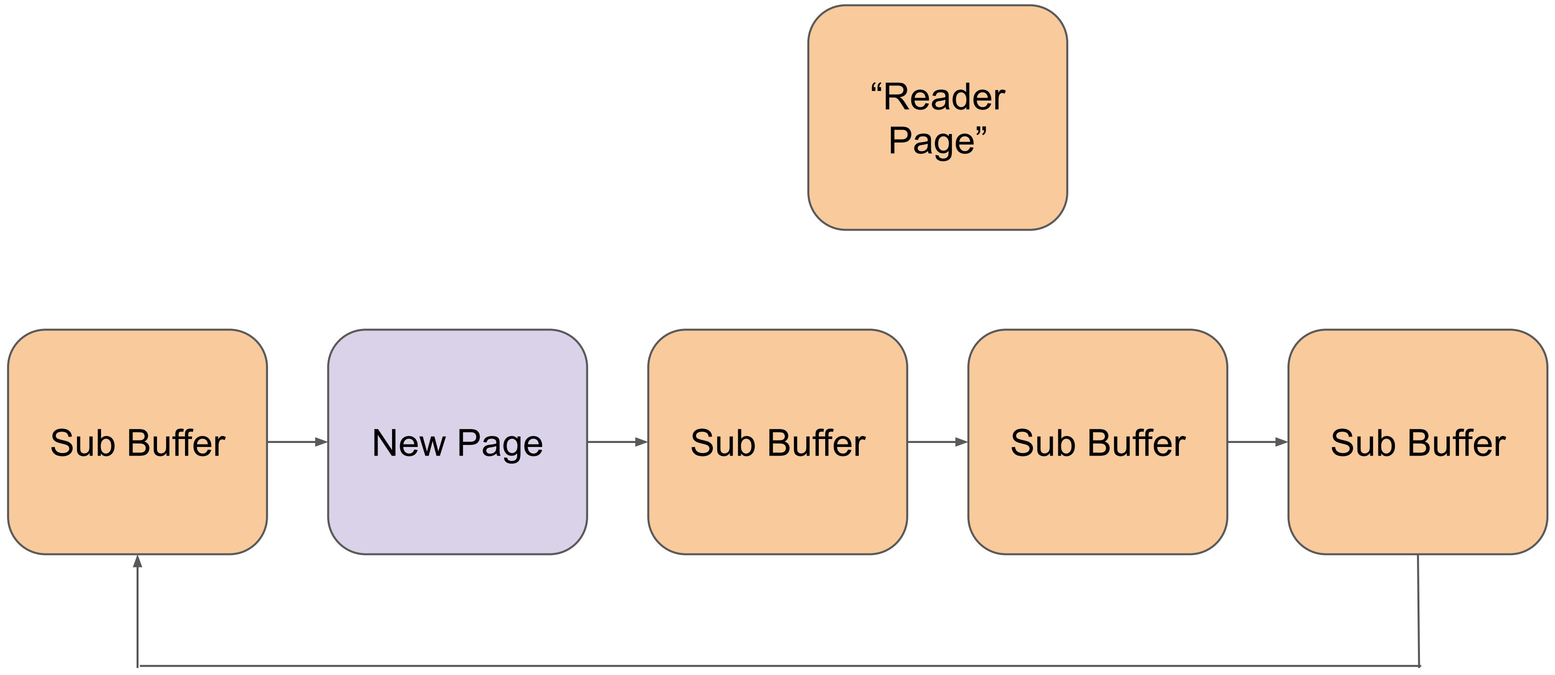
Ftrace ring buffer and splice



Ftrace ring buffer and splice



Ftrace ring buffer and splice



The “Reader Page”

- Reader page is the sub buffer that will not be written to anymore

The “Reader Page”

- Reader page is the sub buffer that will not be written to anymore
- Can be used for anything the reader wants it for

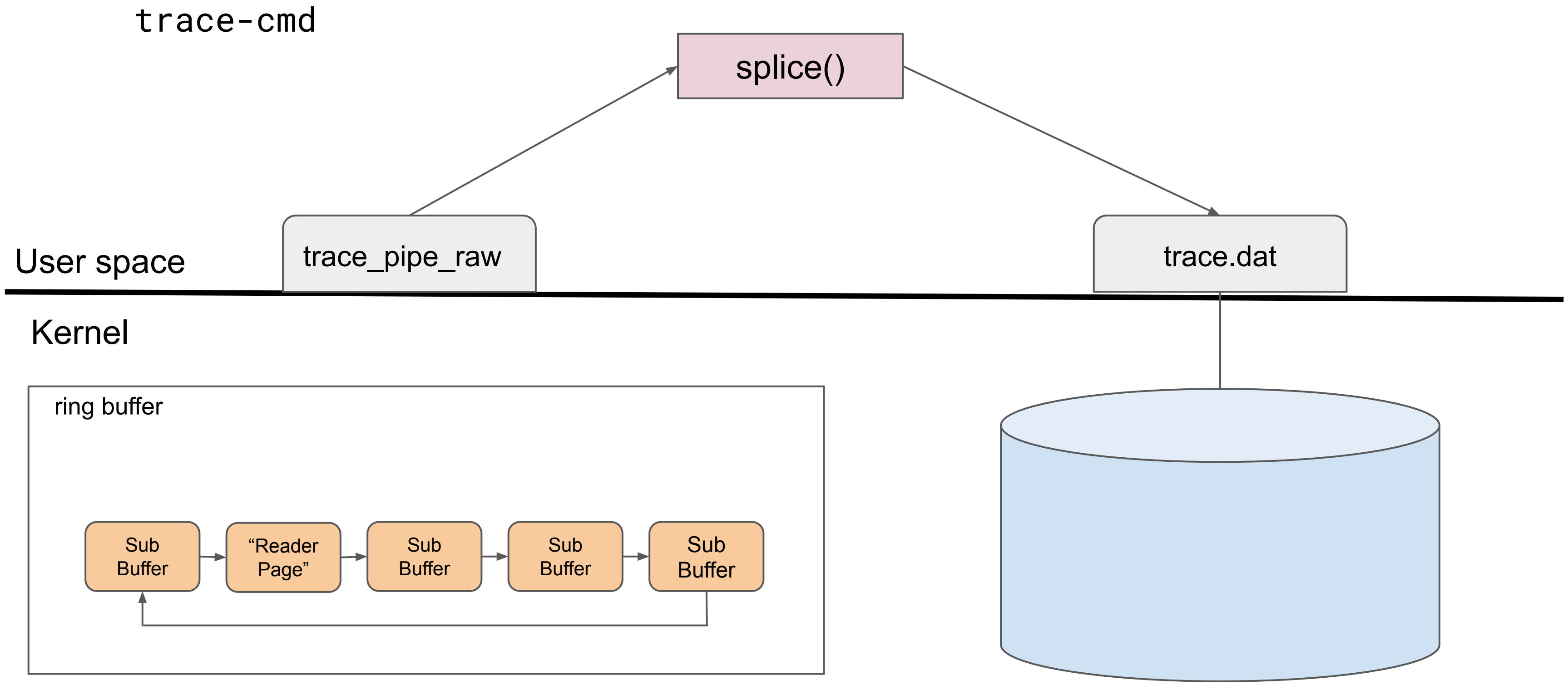
The “Reader Page”

- Reader page is the sub buffer that will not be written to anymore
- Can be used for anything the reader wants it for
- Send it to the disk (into a file)

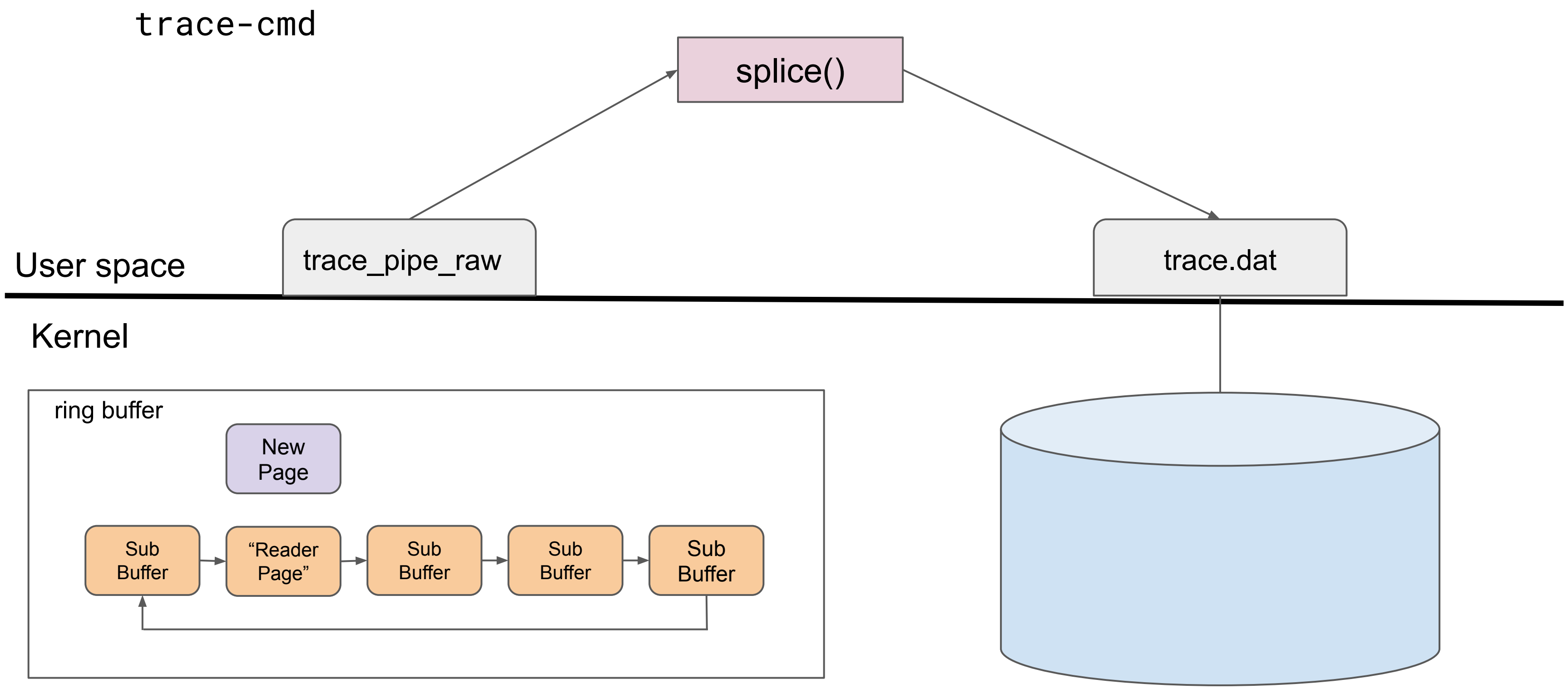
The “Reader Page”

- Reader page is the sub buffer that will not be written to anymore
- Can be used for anything the reader wants it for
- Send it to the disk (into a file)
- Send it over a socket (over the network)

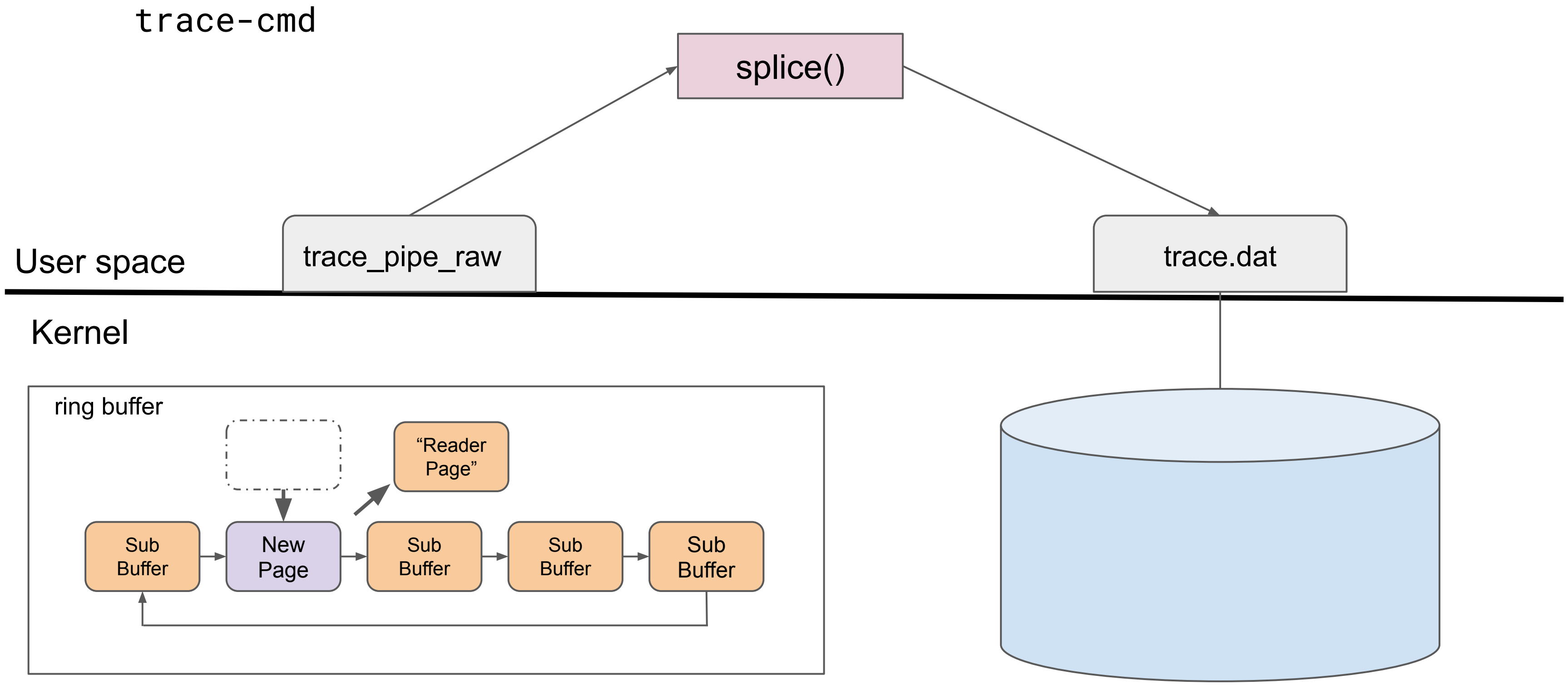
trace-cmd, ftrace and splice



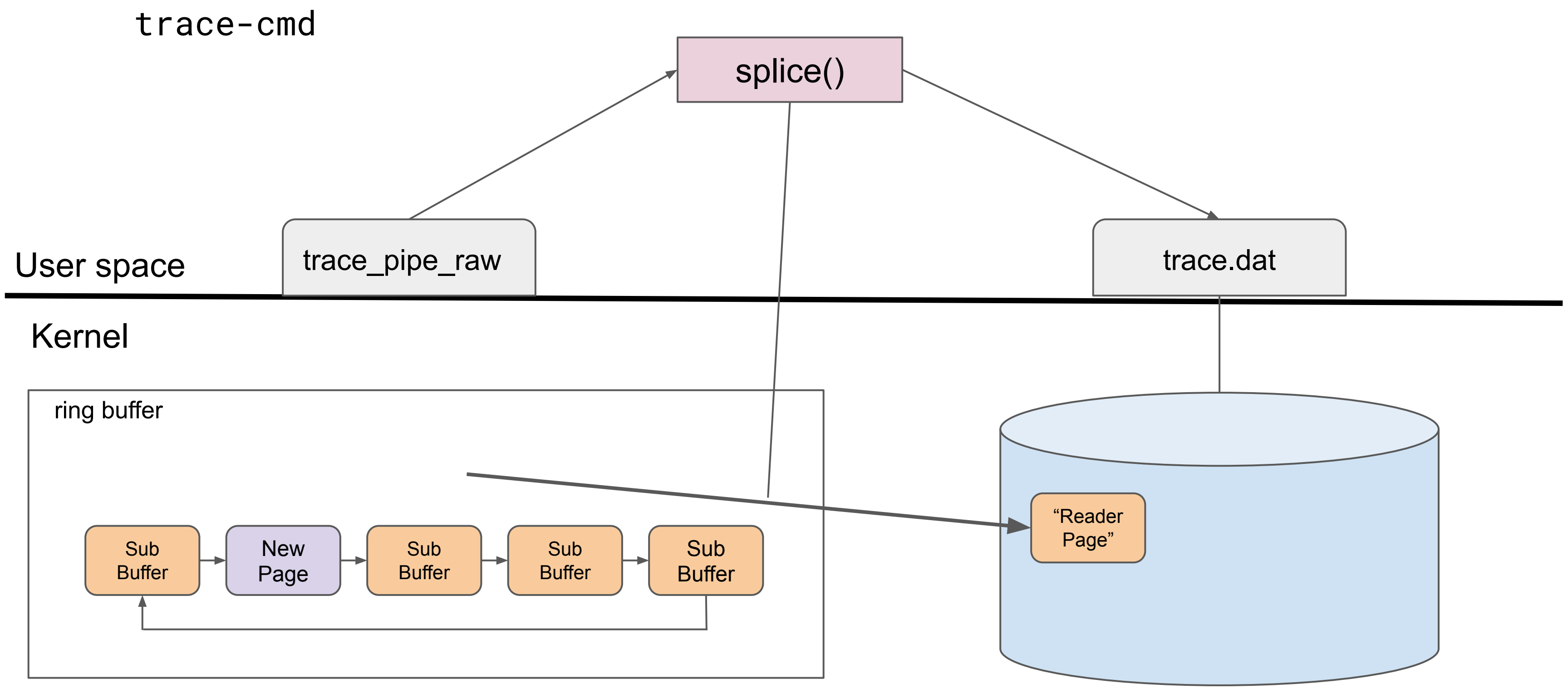
trace-cmd, ftrace and splice



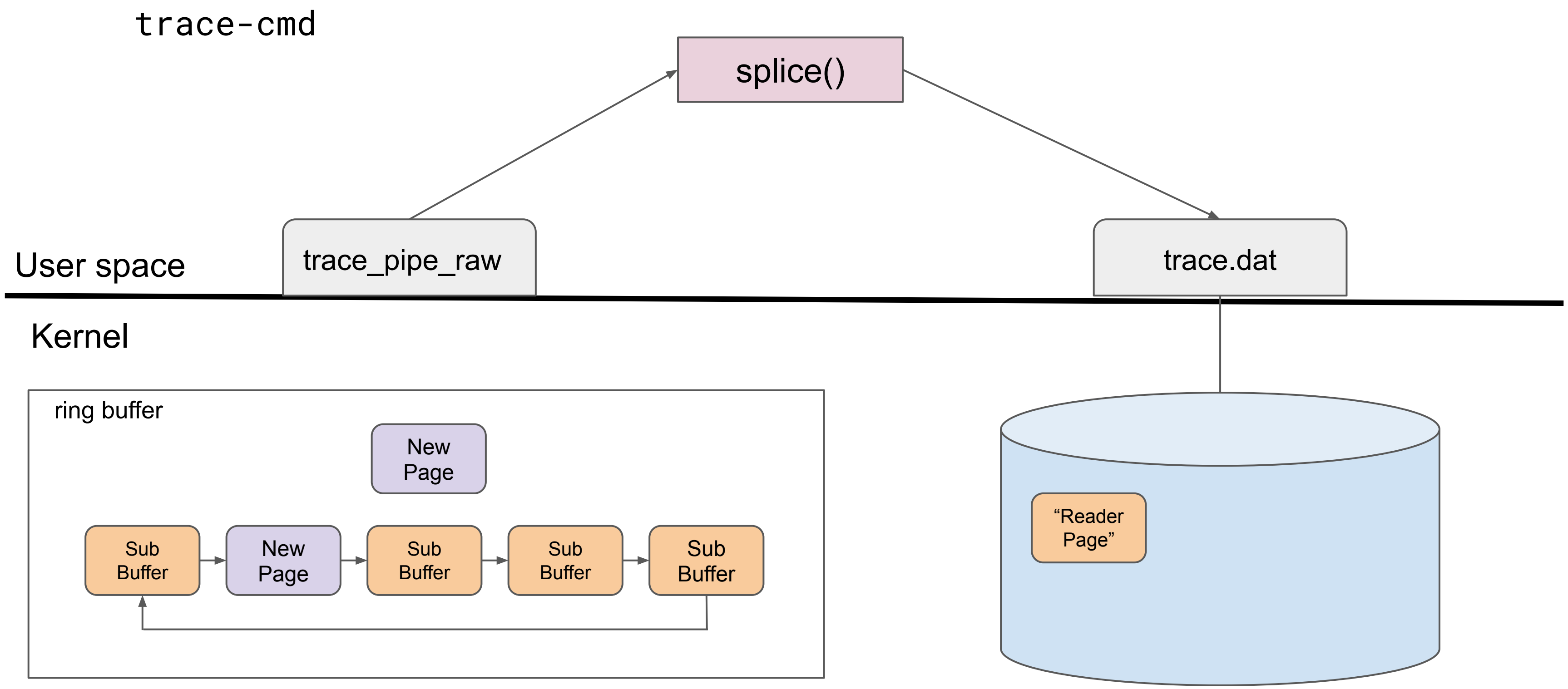
trace-cmd, ftrace and splice



trace-cmd, ftrace and splice



trace-cmd, ftrace and splice



Tracing embedded boards

- Having a trace.dat file is advantageous

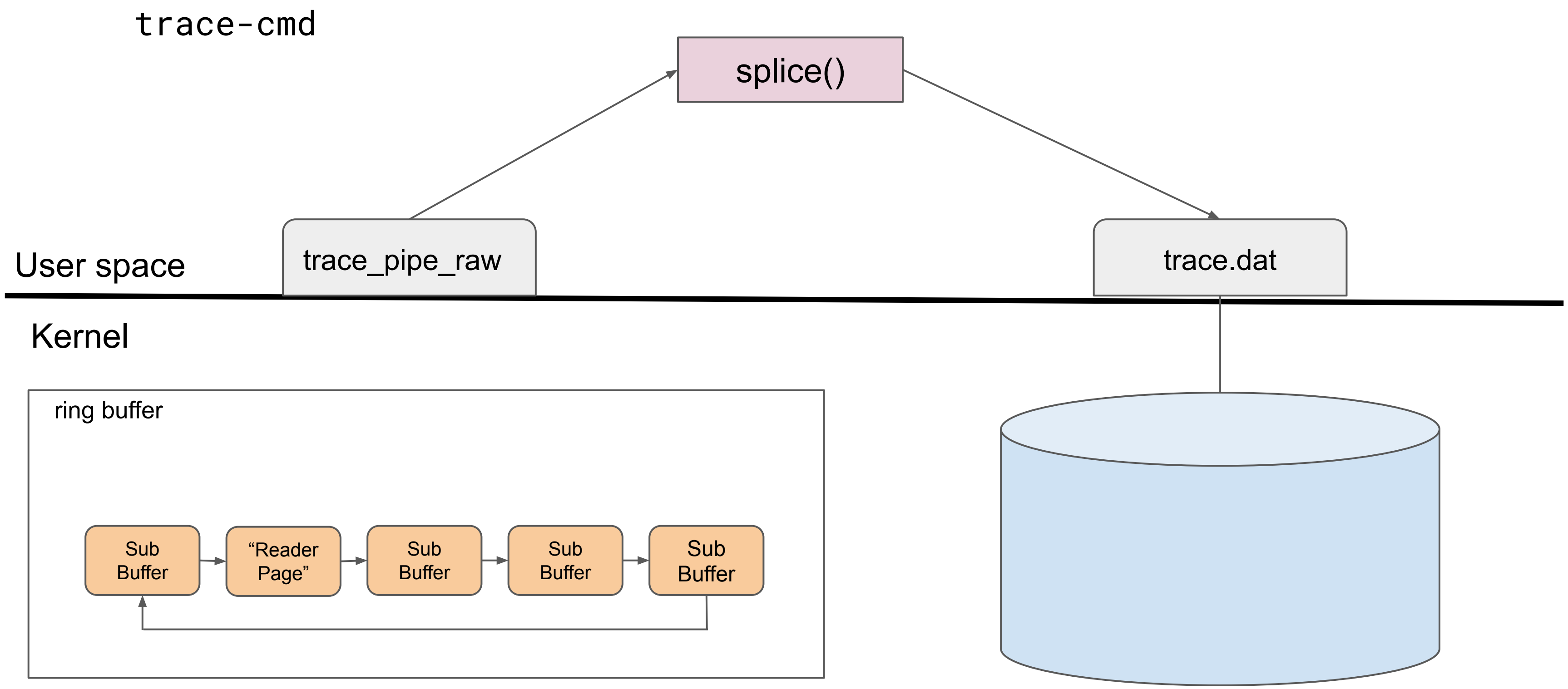
Tracing embedded boards

- Having a trace.dat file is advantageous
- But what happens when there’s little or no disk space

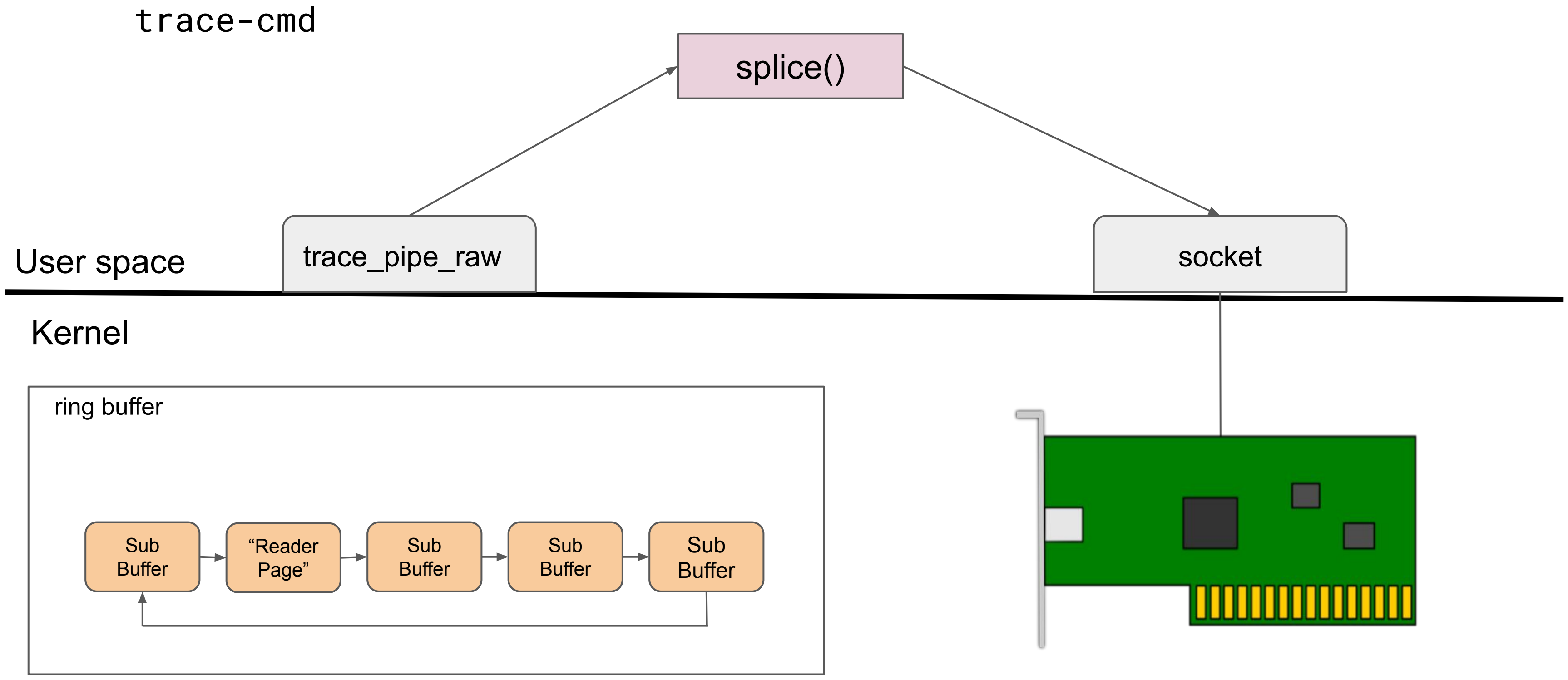
Tracing embedded boards

- Having a trace.dat file is advantageous
- But what happens when there’s little or no disk space
- Need to send it someplace else.

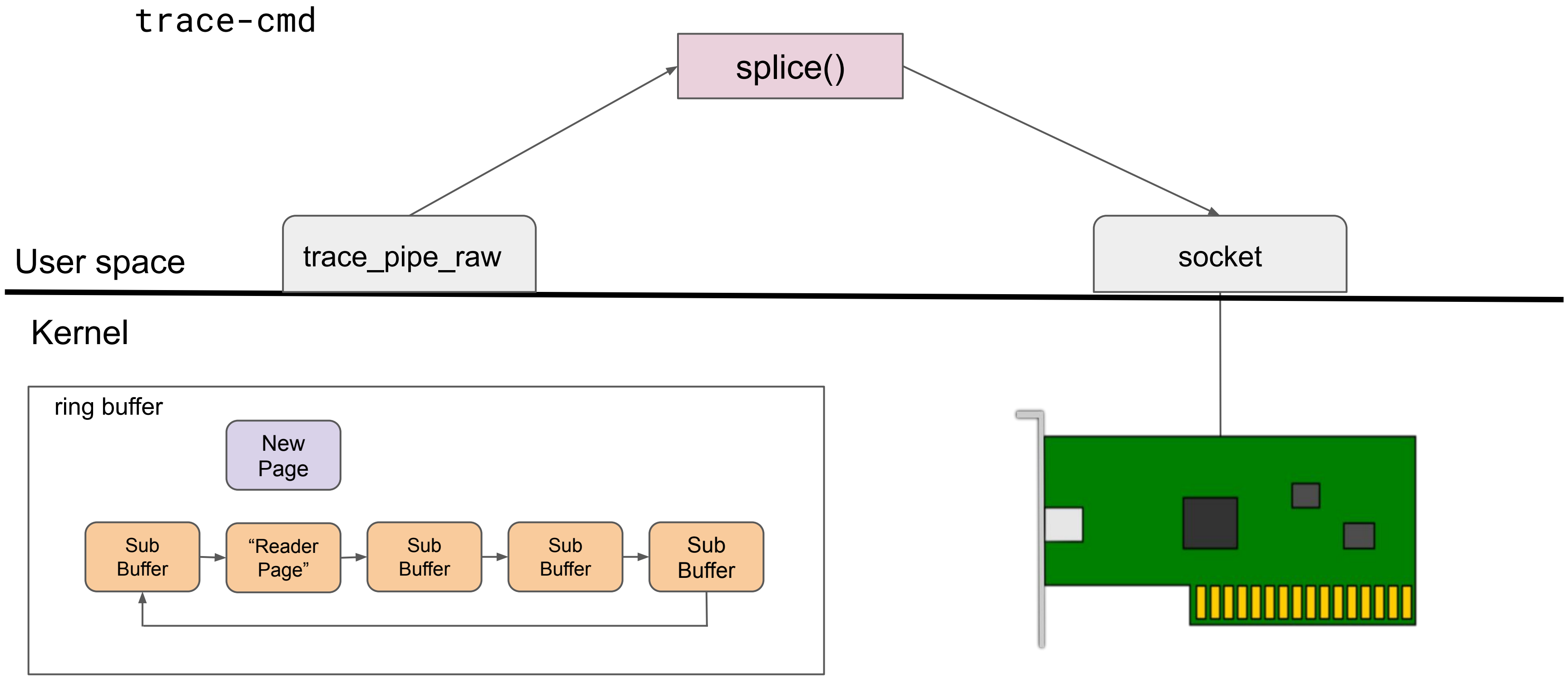
trace-cmd, ftrace and splice



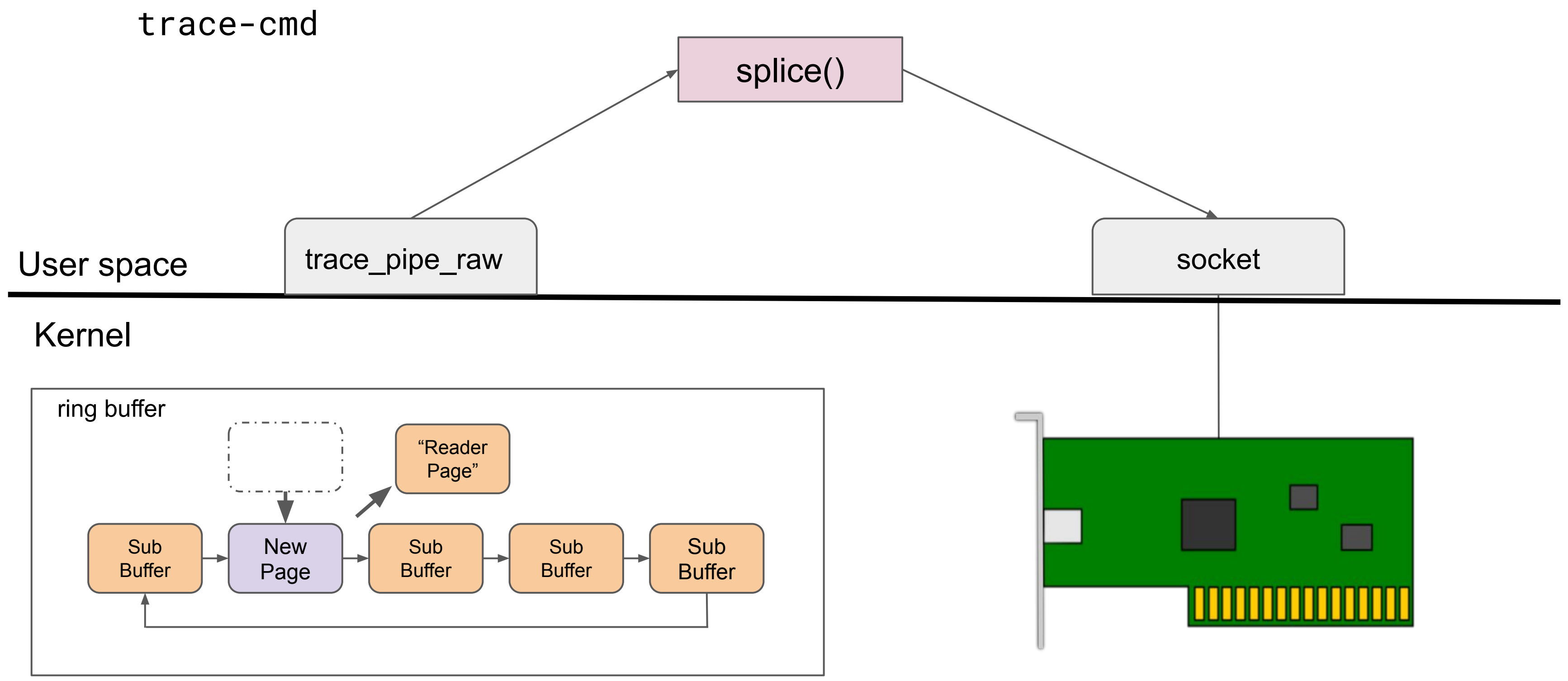
trace-cmd, ftrace and splice, and the network



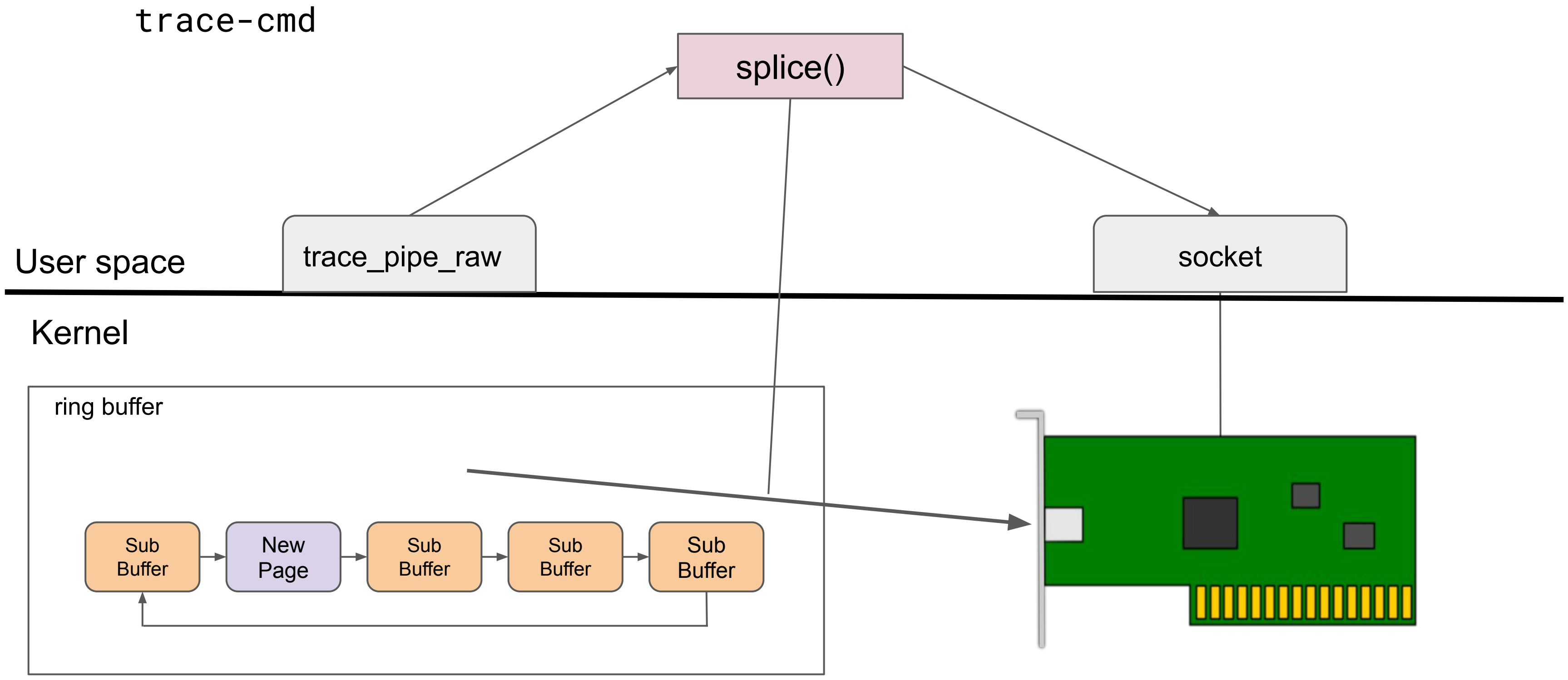
trace-cmd, ftrace and splice, and the network



trace-cmd, ftrace and splice, and the network



trace-cmd, ftrace and splice, and the network



Why do I care?

Why do I care?

- I use to be an embedded developer

Why do I care?

- I use to be an embedded developer
- I was debugging a small ARM board

Why do I care?

- I use to be an embedded developer
- I was debugging a small ARM board

Why do I care?

- I use to be an embedded developer
- I was debugging a small ARM board
- It did not have much disk space

Why do I care?

- I use to be an embedded developer
- I was debugging a small ARM board
- It did not have much disk space
- Decide to record over the network

Why do I care?

- I use to be an embedded developer
- I was debugging a small ARM board
- It did not have much disk space
- Decide to record over the network
- Created `trace-cmd listen`

trace-cmd listen

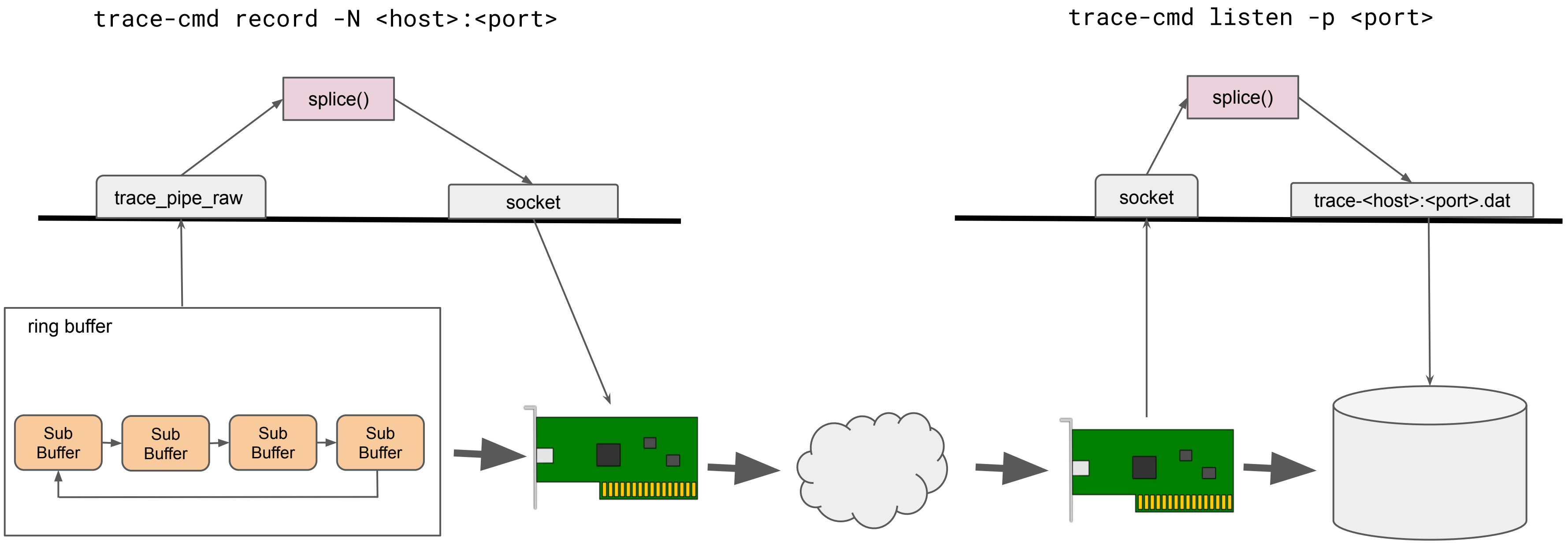
On the server:

```
$ trace-cmd listen -p 22222
```

On the target:

```
# trace-cmd record -N <host>:2222 -e sched -e irq -e timer
```

trace-cmd, ftrace and splice, and the network



trace-cmd listen

- Runs on a server (x86 Workstation, or whatever)

trace-cmd listen

- Runs on a server (x86 Workstation, or whatever)
- Takes connections from anywhere

trace-cmd listen

- Runs on a server (x86 Workstation, or whatever)
- Takes connections from anywhere
  - Not secure (but has limited vulnerabilities)
  - Could use iptables to secure it more
  - Do not leave the port open to the Internet!

trace-cmd listen

- Runs on a server (x86 Workstation, or whatever)
- Takes connections from anywhere
    - Not secure (but has limited vulnerabilities)
    - Could use iptables to secure it more
    - Do not leave the port open to the Internet!
- Requires recording from the target

trace-cmd listen

- Runs on a server (x86 Workstation, or whatever)
- Takes connections from anywhere
  - Not secure (but has limited vulnerabilities)
  - Could use iptables to secure it more
  - Do not leave the port open to the Internet!
- Requires recording from the target
  - Need ssh console

trace-cmd listen

- Runs on a server (x86 Workstation, or whatever)
- Takes connections from anywhere
  - Not secure (but has limited vulnerabilities)
  - Could use iptables to secure it more
  - Do not leave the port open to the Internet!
- Requires recording from the target
  - Need ssh console
  - Does not synchronize any events with the workstation (listener)

Introducing the “Agent”

Introducing the “Agent”

- Was introduced to trace between host and guest

Introducing the “Agent”

- Was introduced to trace between host and guest
- Allows remote enabling of tracing

Introducing the “Agent”

- Was introduced to trace between host and guest
- Allows remote enabling of tracing
- Synchronizes time stamps with the host!

Introducing the “Agent”

- Was introduced to trace between host and guest
- Allows remote enabling of tracing
- Synchronizes time stamps with the host!
- Sends data over a socket or FIFO

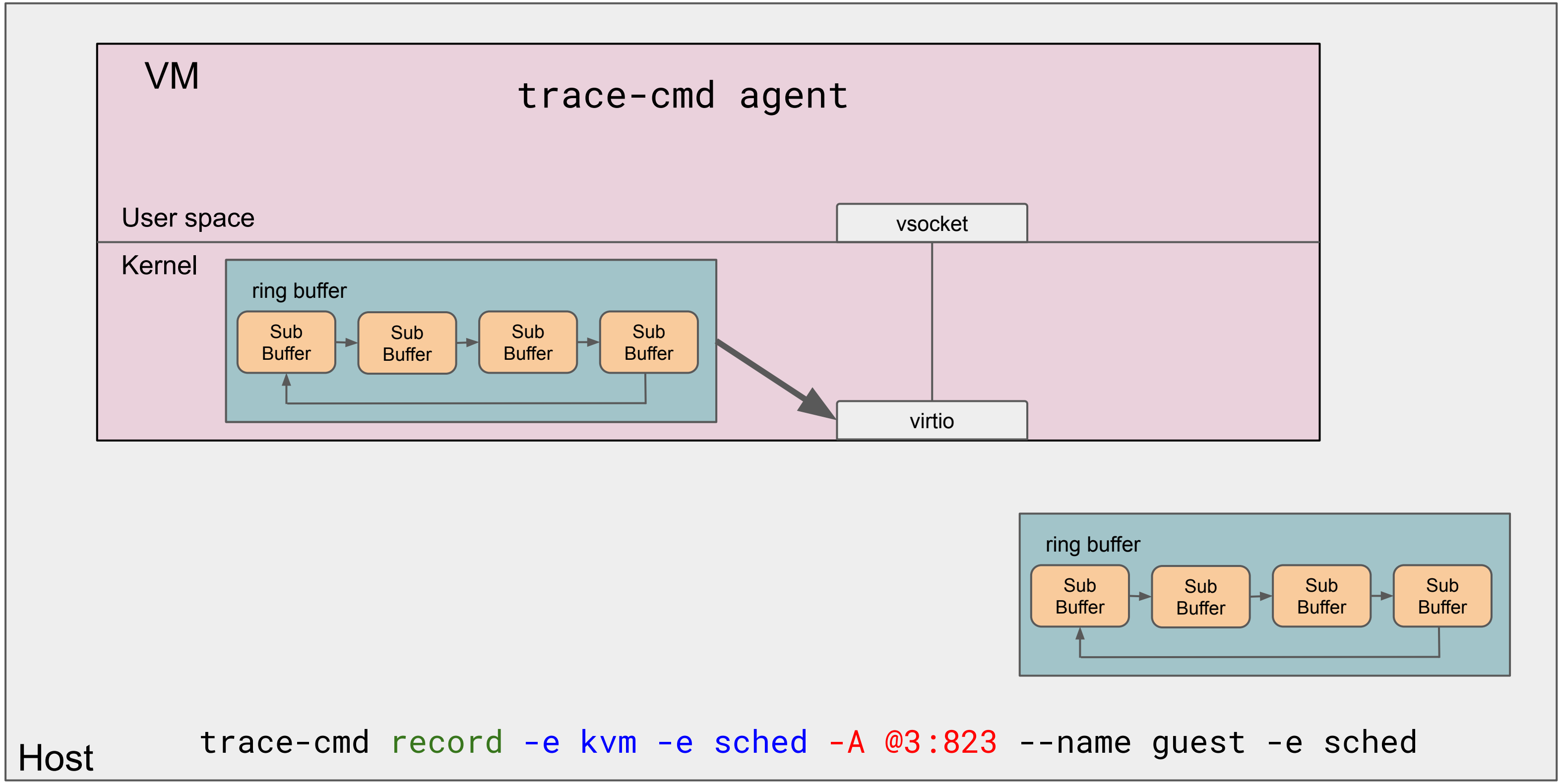
Introducing the “Agent”

- Was introduced to trace between host and guest
- Allows remote enabling of tracing
- Synchronizes time stamps with the host!
- Sends data over a socket or FIFO
  - vsockets

Introducing the “Agent”

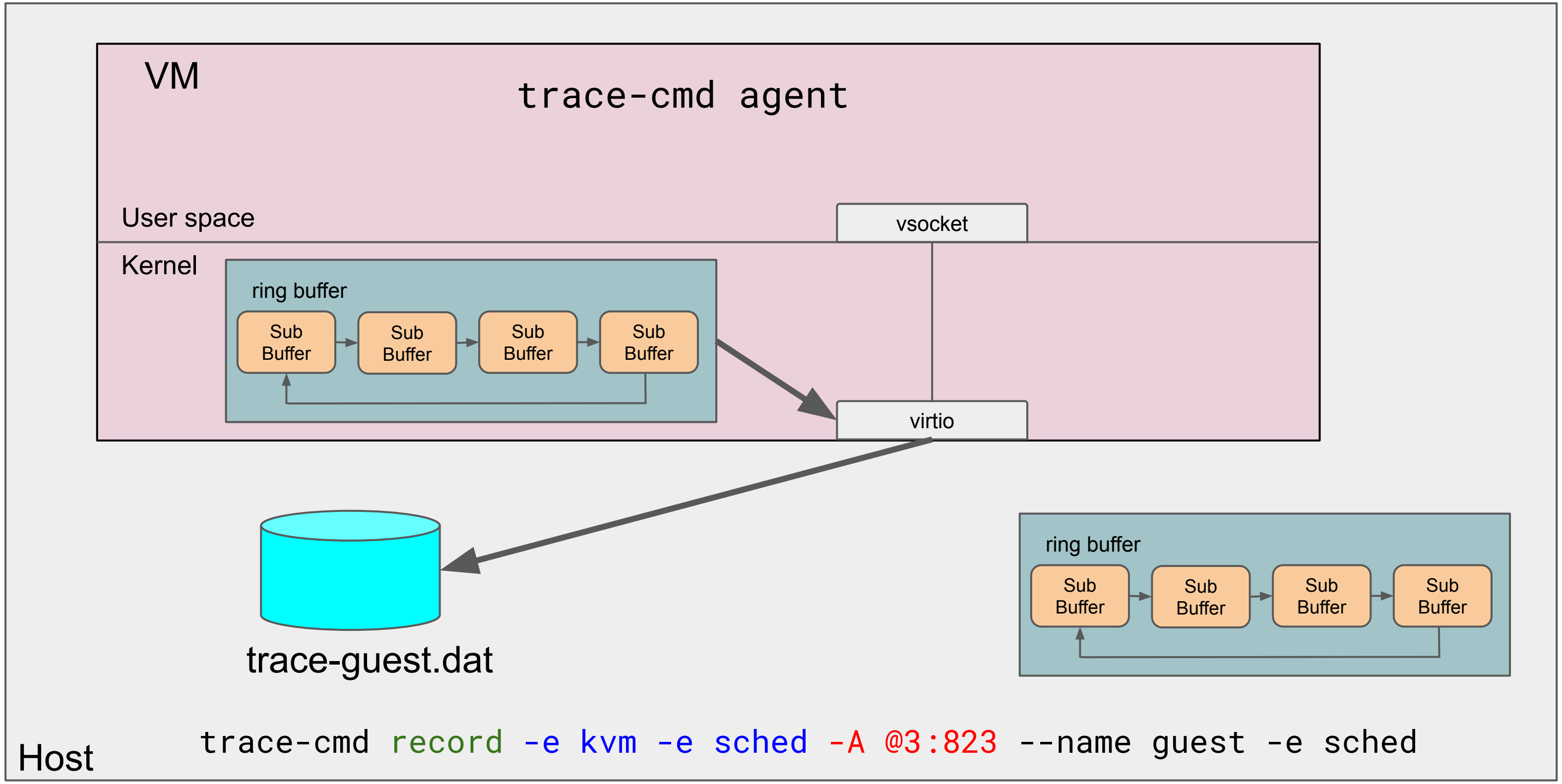
- Was introduced to trace between host and guest
- Allows remote enabling of tracing
- Synchronizes time stamps with the host!
- Sends data over a socket or FIFO
  - vsockets
  - Now can use network sockets!

trace-cmd agent



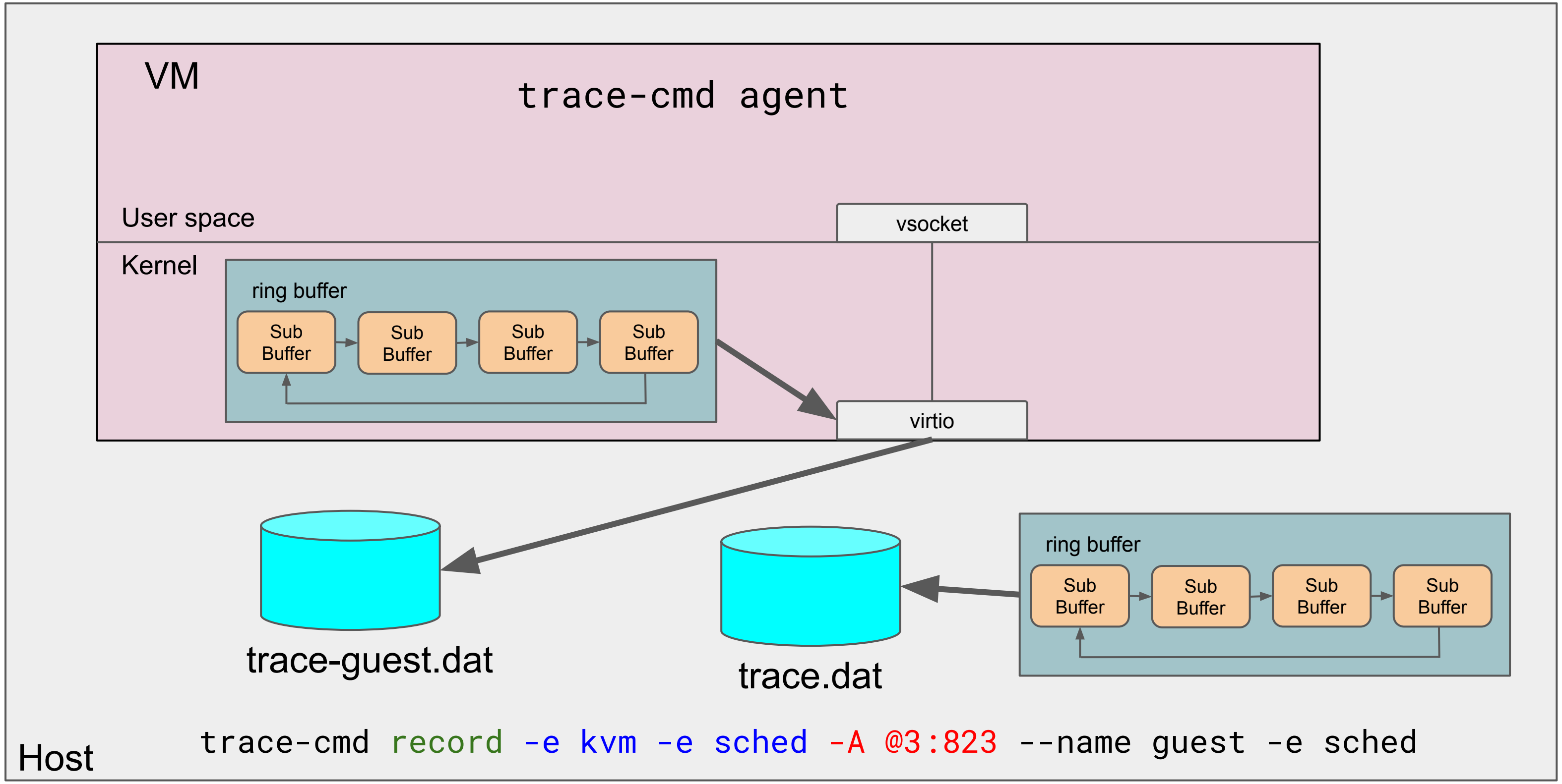
```
trace-cmd record -e kvm -e sched -A @3:823 --name guest -e sched
```

trace-cmd agent



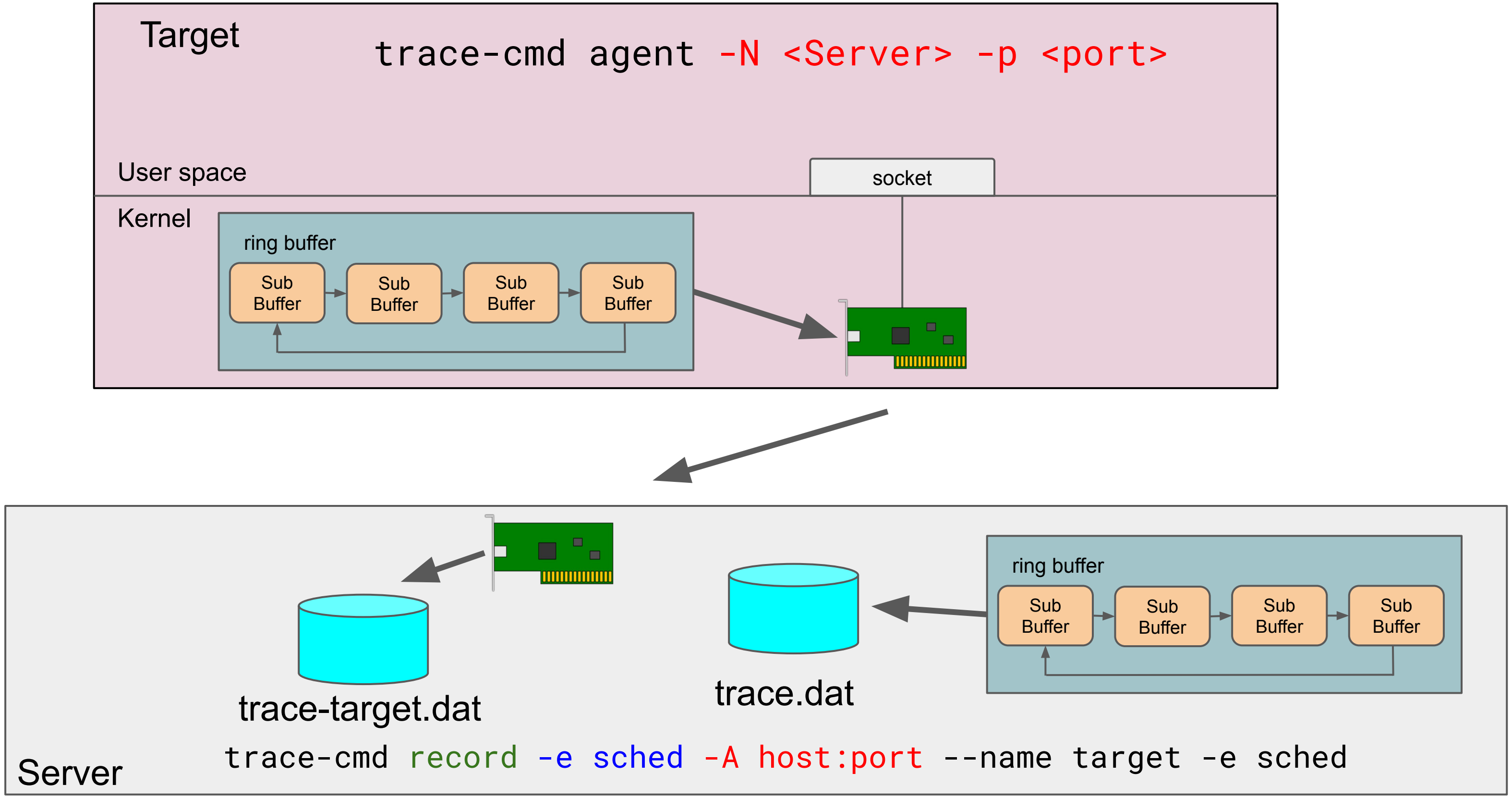
```
trace-cmd record -e kvm -e sched -A @3:823 --name guest -e sched
```

trace-cmd agent



```
trace-cmd record -e kvm -e sched -A @3:823 --name guest -e sched
```

trace-cmd agent over network



```
trace-cmd agent -N <Server> -p <port>
```

```
trace-cmd record -e sched -A host:port --name target -e sched
```

Using the agent over the network

- Allows a remote server to control the board

Using the agent over the network

- Allows a remote server to control the board
- Must be a “trusted” server

Using the agent over the network

- Allows a remote server to control the board
- Must be a “trusted” server
  - Anyone on the server can control the board

Using the agent over the network

- Allows a remote server to control the board
- Must be a “trusted” server
    - Anyone on the server can control the board
- Allows remote enabling of tracing

Using the agent over the network

- Allows a remote server to control the board
- Must be a “trusted” server
    - Anyone on the server can control the board
- Allows remote enabling of tracing
- Still has synchronization between events
    - P2P time synchronization protocol

Thank you!