Common Tracing Platform

Let’s put our brains together

Tzvetomir Stoyanov & Steven Rostedt
10/25/2018

vmware

First, libtraceevent

- The library to parse the trace events file
  - This file explains how to read the trace binary data
  - Allows events to change and not break tools
- Used by trace-cmd, perf, PowerTop, mceutils
  - And more (people keep asking me for this)
- Written using several “generic” names
  - event_format
  - record
  - event_field

The event format file

```
# cat /sys/kernel/debug/tracing/events/sched/sched_switch/format
name: sched_switch
ID: 311
format:
        field:unsigned short common_type;   offset:0;         size:2; signed:0;
        field:unsigned char common_flags;   offset:2;         size:1; signed:0;
        field:unsigned char common_preempt_count;         offset:3;         size:1; signed:0;
        field:int common_pid; offset:4;         size:4; signed:1;

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

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

The event format file

```
# cat /sys/kernel/debug/tracing/events/sched/sched_switch/format
name: sched_switch
ID: 311
format:
        field:unsigned short common_type; offset:0;       size:2; signed:0;
        field:unsigned char common_flags; offset:2;       size:1; signed:0;
        field:unsigned char common_preempt_count;     offset:3;       size:1; signed:0;
        field:int common_pid;       offset:4;       size:4; signed:1;

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

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

The event format file

 # cat /sys/kernel/debug/tracing/events/sched/sched_switch/format

```
name: sched_switch
ID: 311
format:
        field:unsigned short common_type;   offset:0;         size:2; signed:0;
        field:unsigned char common_flags;   offset:2;         size:1; signed:0;
        field:unsigned char common_preempt_count;         offset:3;         size:1; signed:0;
        field:int common_pid; offset:4;         size:4; signed:1;

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

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

The event format file

 # cat /sys/kernel/debug/tracing/events/sched/sched_switch/format

```
name: sched_switch
ID: 311
format:
        field:unsigned short common_type;   offset:0;         size:2; signed:0;
        field:unsigned char common_flags;   offset:2;         size:1; signed:0;
        field:unsigned char common_preempt_count;         offset:3;         size:1; signed:0;
        field:int common_pid; offset:4;         size:4; signed:1;

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

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

libtraceevent parsing

- Currently was written by hand
  - Makes it hard to maintain
  - Harder to make correct
    - A bug was just reported yesterday
  - Makes it hard to modify

libtraceevent parsing

- Working to replace it using Flex and Bison
  - Easier to maintain
  - Easier to modify
  - Easier to understand
  - Easier to have others make changes

libtraceevent parsing

- Working to replace it using Flex and Bison
  - Easier to maintain
  - Easier to modify
  - Easier to understand
  - Easier to have others make changes
  - Harder to get right the first time
  - Harder to deal with the “strange format”
  - Pretty much a full C parser (for the print fmt)

libtraceevent parsing

- Working to replace it using Flex and Bison
  - Easier to maintain
  - Easier to modify
  - Easier to understand
  - Easier to have others make changes
  - Harder to get right the first time
  - Harder to deal with the “strange format”
  - Pretty much a full C parser (for the print fmt)

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

libtraceevent name space

- Getting it into a library form
  - Renamed “pevent” to “tep” (Trace Event Parser)
    - “tep” defines the library’s name space
  - record -> tep_record
  - event_format -> tep_event
  - event_field -> tep_event_field
  - All the visible functions and structures now start with “tep_”

Linking to libtraceevent

- autoconfig
  - pkg-config
    - OK to add to tools/lib/traceevent?

libtraceevent man pages

LIBTRACEEVENT(3) trace-cmd Manual LIBTRACEEVENT(3)

NAME

libtraceevent - Linux kernel trace event library

SYNOPSIS

#include <event-parse.h>

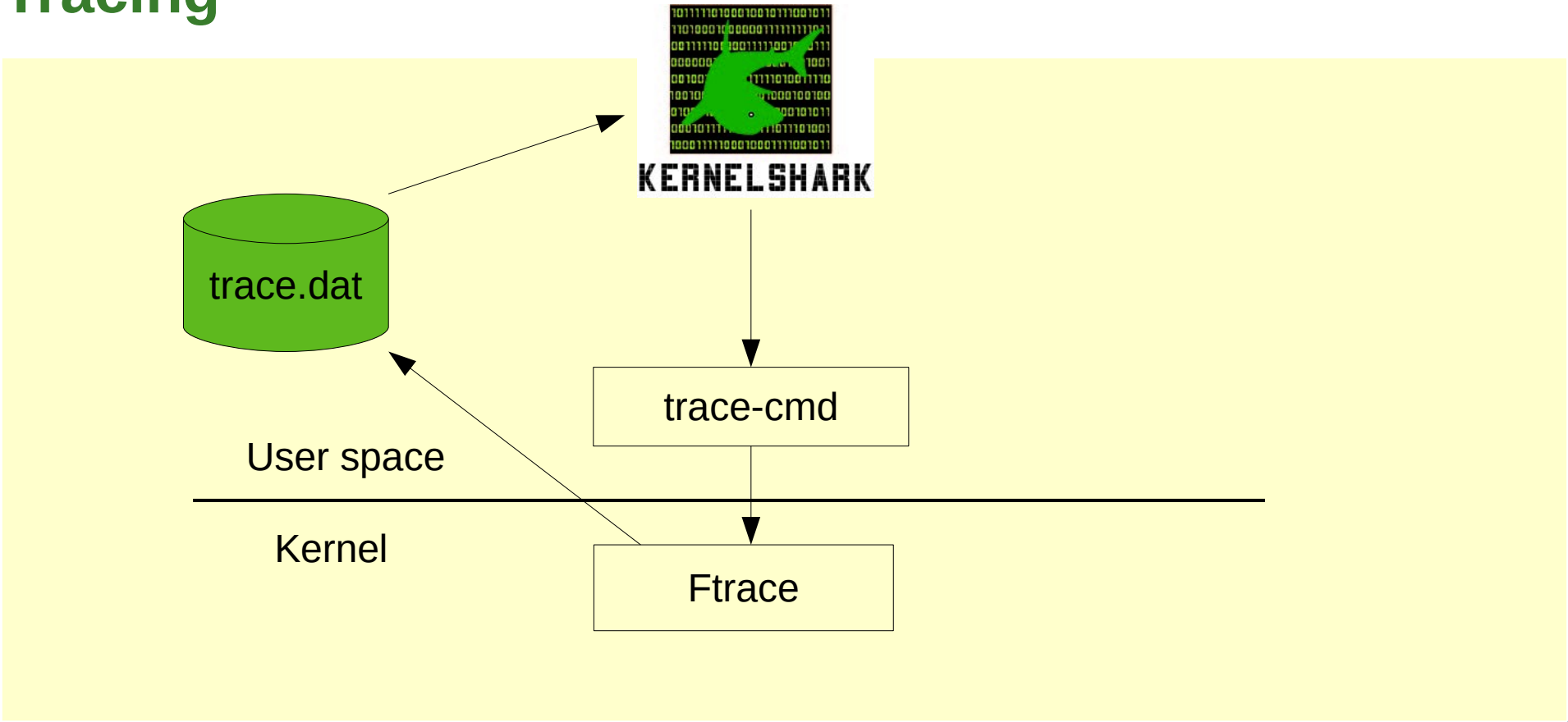
Management of tep handler data structure and access of its members:

```
struct tep_handle *tep_alloc(void);
void tep_free(struct tep_handle *pevent);
void tep_ref(struct tep_handle *pevent);
void tep_unref(struct tep_handle *pevent);
int tep_ref_get(struct tep_handle *pevent);
void tep_set_flag(struct tep_handle *tep, int flag);
int tep_get_cpus(struct tep_handle *pevent);
void tep_set_cpus(struct tep_handle *pevent, int cpus);
int tep_get_long_size(strucqt tep_handle *pevent);
void tep_set_long_size(struct tep_handle *pevent, int long_size);
int tep_get_page_size(struct tep_handle *pevent);
void tep_set_page_size(struct tep_handle *pevent, int page_size);
int tep_is_file_bigendian(struct tep_handle *pevent);
void tep_set_file_bigendian(struct tep_handle *pevent, enum tep_endian endian);
int tep_is_host_bigendian(struct tep_handle *pevent);
void tep_set_host_bigendian(struct tep_handle *pevent, enum tep_endian endian);
int tep_is_latency_format(struct tep_handle *pevent);
void tep_set_latency_format(struct tep_handle *pevent, int lat);
int tep_get_header_page_size(struct tep_handle *pevent);
int tep_register_trace_clock(struct tep_handle *pevent, const char *trace_clock);
int tep_register_function(struct tep_handle *pevent, char *name, unsigned long long addr, char *mod);
```

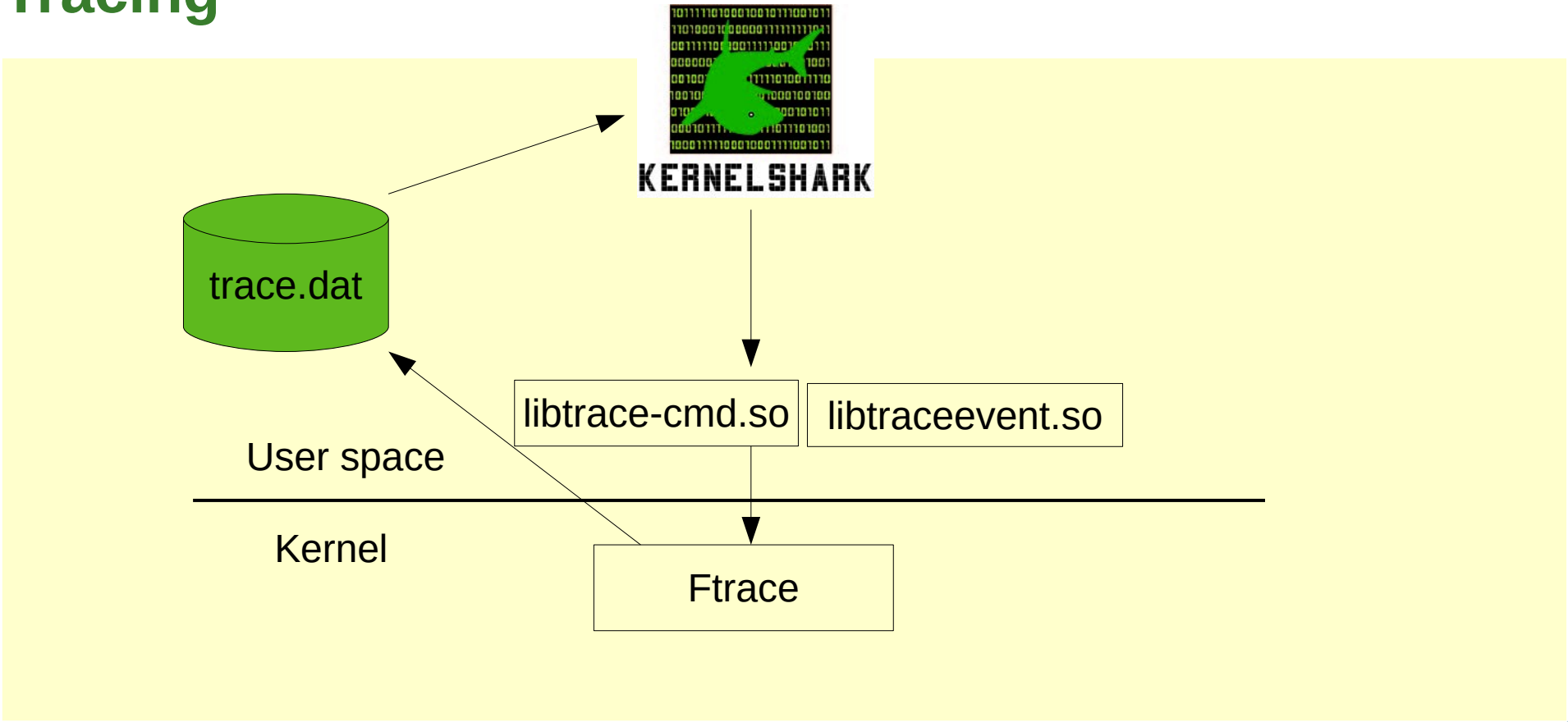
libtraceevent man pages

- More than 100 functions
  - Michael Kerrisk suggests to split up each one
- Store them in tools/lib/traceevent/Documentation
- Need to vet each one
  - Once released they shall be written in stone!
  - Must be backward and forward compatible

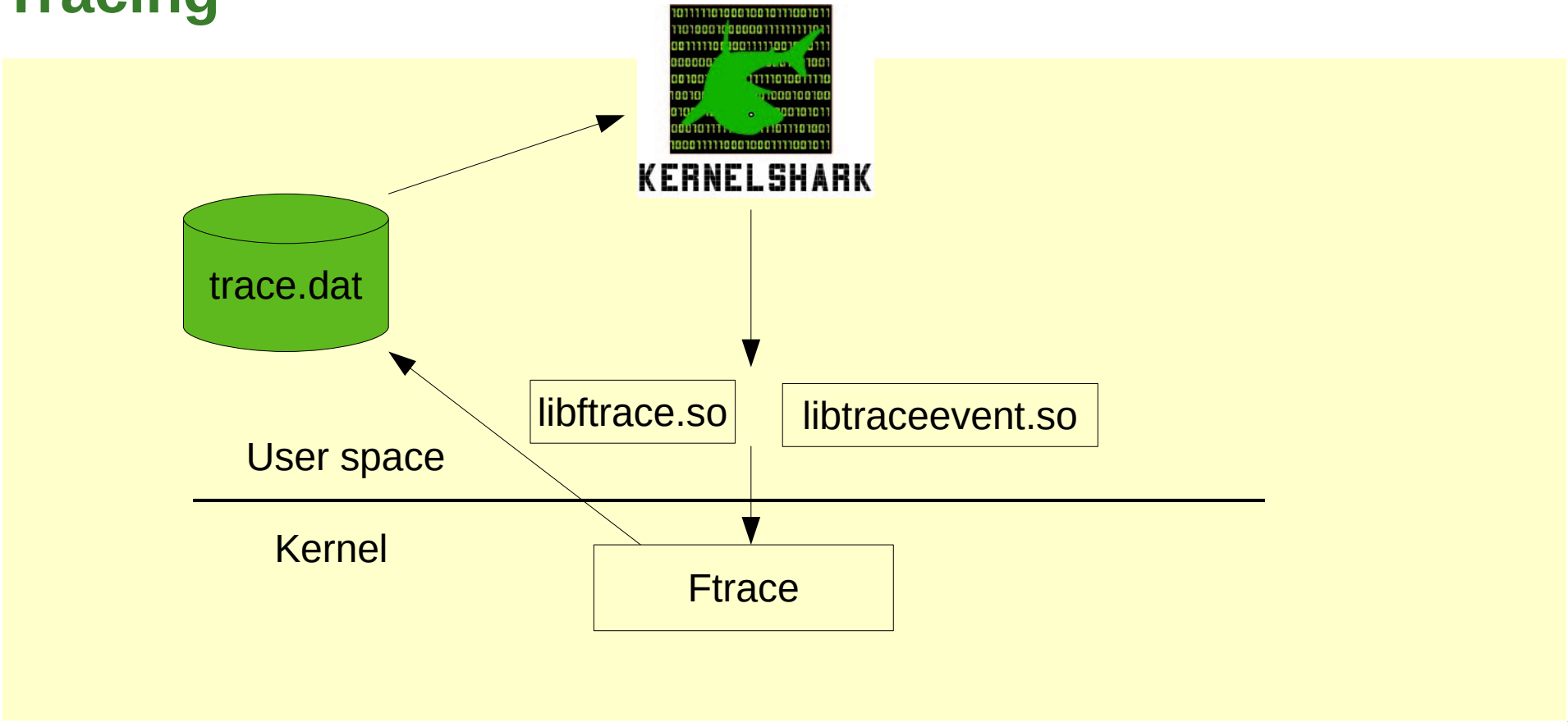
Tracing



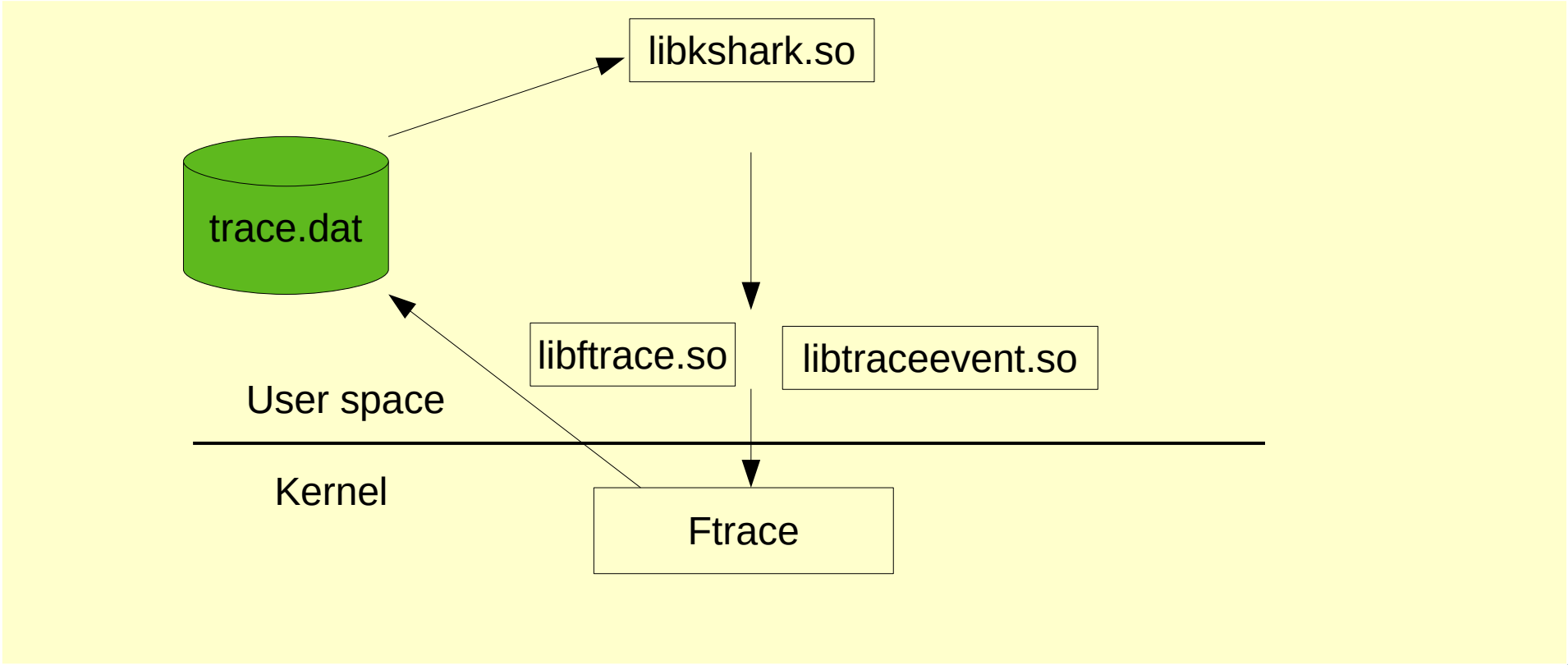
Tracing


KERNELSHARK
trace.dat
libtrace-cmd.so
libtraceevent.so
User space
Kernel
Ftrace

Tracing

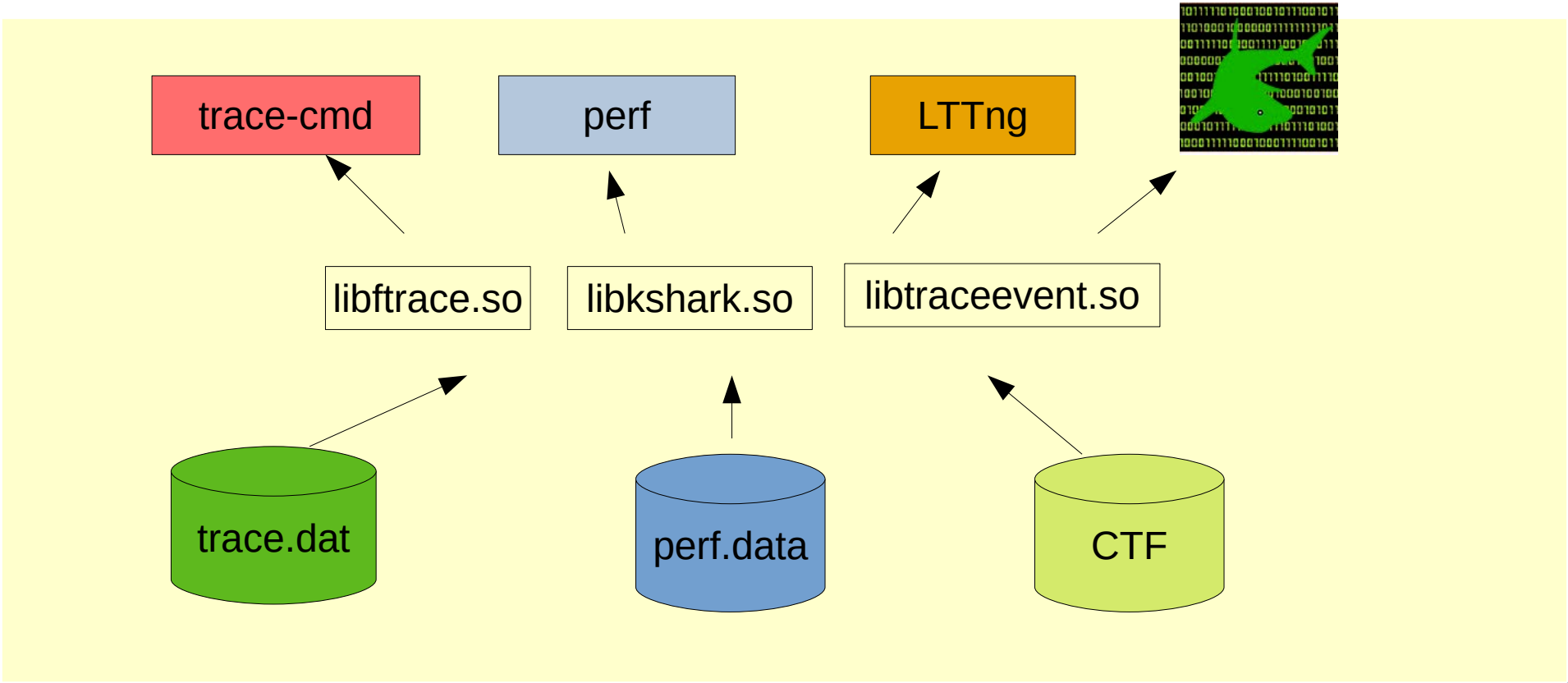

KERNELSHARK
trace.dat
libftrace.so
libtraceevent.so
User space
Kernel
Ftrace

Tracing

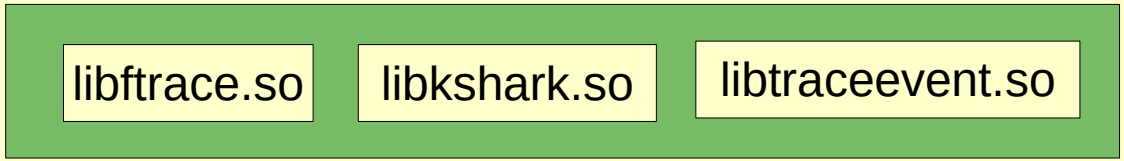

libkshark.so
trace.dat
libftrace.so
libtraceevent.so
User space
Kernel
Ftrace

Tracing Platform

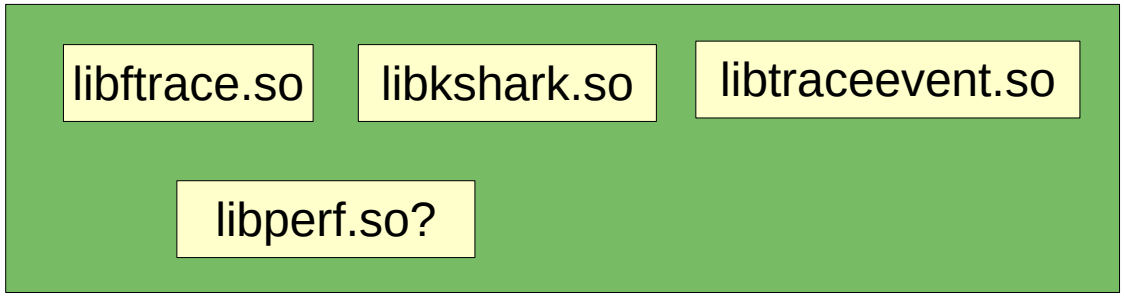
Tracing Platform



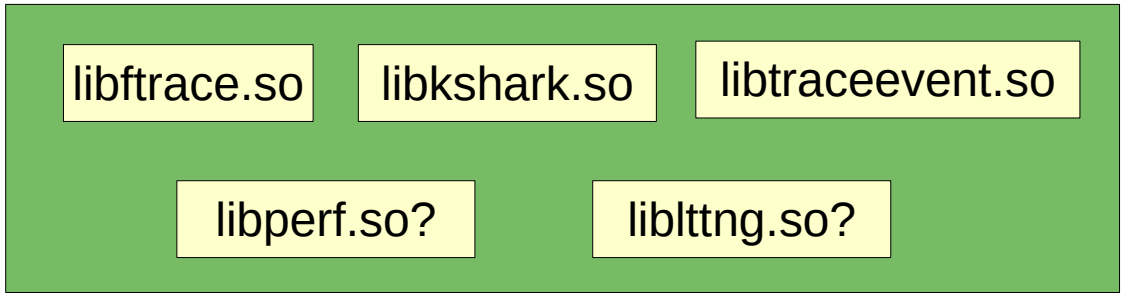
Tracing Platform



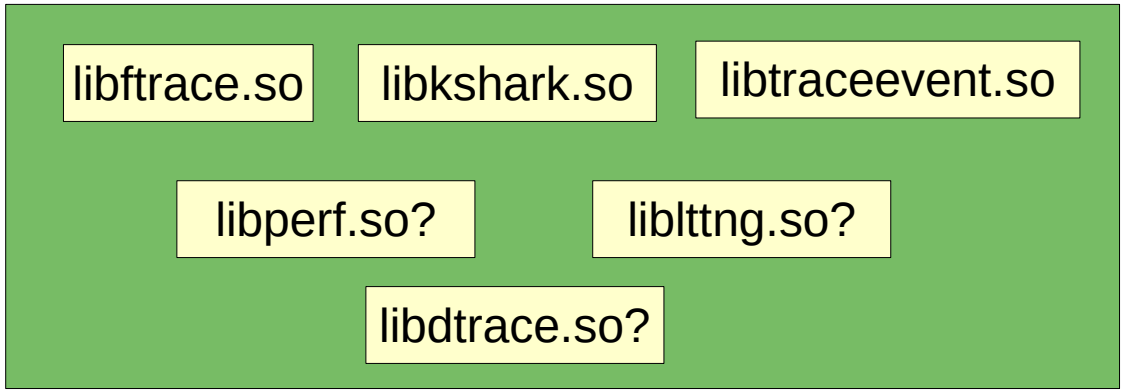
Tracing Platform



Tracing Platform



Tracing Platform



Discussion!