Formal sound verification of Linux's USB BP keyboard driver

Willem Penninckx
Jan Tobias Mühlberg
Jan Smans
Bart Jacobs
Frank Piessens

Table Of Contents

Table Of Contents

Formal sound verification

~~Bug hunting~~

If “green bar”, then
verified property always holds

Check properties:
- Never crashes
- No race-condition
- API rules

- Real-world software ~~toy~~
- Unbounded number of threads
- Unbounded number of keyboards

Linux’s USB BP keyboard driver

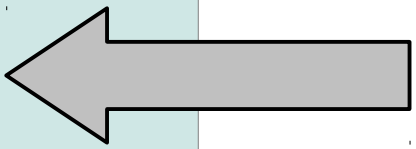
Table Of Contents

usbkbd.c

```
/*@
 * PreCond
 * PostCond
@*/
void fun1() {
    c_code;
    //@ ghostcode
    c_code;
    c_code;
}
```

input.h

```
/*@
 * preCond
 * postCond
@*/
void input_register();
```



Formal API specs

usb.h

```
/*@
 * preCond
 * postCond
@*/
void usb_kill_urb();

/*@
 * (ghost code)
@*/
```

usb_core.c

```
void
usb_kill_urb() {
    c_code;
    c_code;
    c_code;
}
```

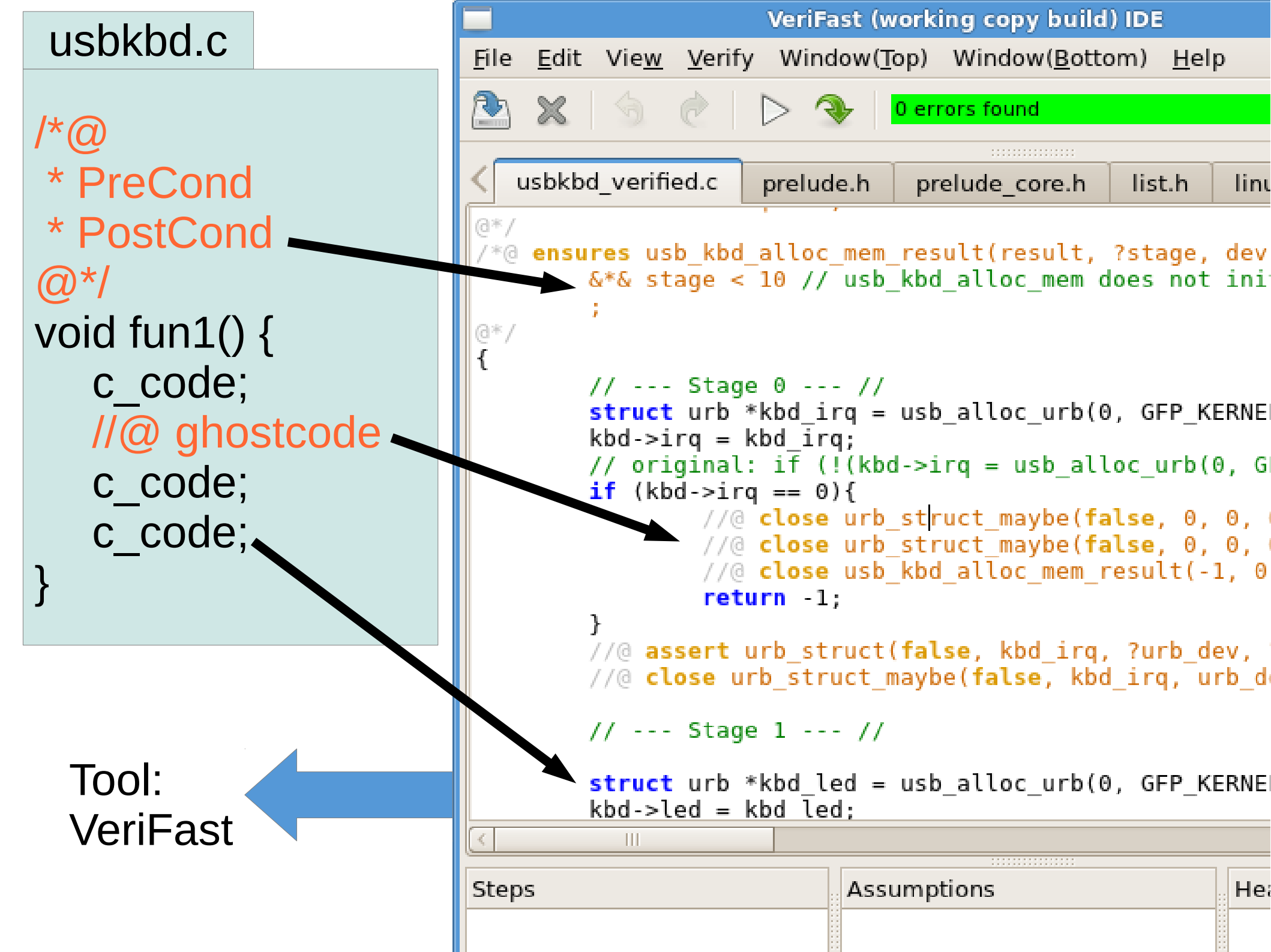

usbkbd.c
/*@
* PreCond
* PostCond
@*/
void fun1() {
c_code;
//@ ghostcode
c_code;
c_code;
}
Tool:
VeriFast
VeriFast (working copy build) IDE
File Edit View Verify Window(Top) Window(Bottom) Help
0 errors found
usbkbd_verified.c
prelude.h
prelude_core.h
list.h
@*/
/*@ ensures usb_kbd_alloc_mem_result(result, ?stage, dev
&*& stage < 10 // usb_kbd_alloc_mem does not ini
;
@*/
{
// --- Stage 0 --- //
struct urb *kbd_irq = usb_alloc_urb(0, GFP_KERNE
kbd->irq = kbd_irq;
// original: if (!(kbd->irq = usb_alloc_urb(0, G
if (kbd->irq == 0){
//@ close urb_struct_maybe(false, 0, 0,
//@ close urb_struct_maybe(false, 0, 0,
//@ close usb_kbd_alloc_mem_result(-1, 0
return -1;
}
//@ assert urb_struct(false, kbd_irq, ?urb_dev,
//@ close urb_struct_maybe(false, kbd_irq, urb_d
// --- Stage 1 --- //
struct urb *kbd_led = usb_alloc_urb(0, GFP_KERNE
kbd->led = kbd_led;
Steps
Assumptions

Table Of Contents

Learned / Conclusions

Possible to combine:

- Soundness
- Unbounded #threads
- Real driver
- API usage rules

Tool speed

~1 second

File	Lines C	Lines annot
usbkbd.c	329	822
API headers	/	769

Bugs found

- Unloading bug
- Synchronization bug

Patches are in Linux 3.3

http://people.cs.kuleuven.be/~willem.penninckx/usbkbd/

Slide to workaround
go-back-after-last-slide-and-miss-final-animation bug