# An Introduction to PS/2 and ps2emu

Stephen Chandler “Lyude” Paul

# What is PS/2?

- Old protocol for mice and keyboard
- Still used in most laptops
- Low bandwidth
- Only allows for two finger multitouch

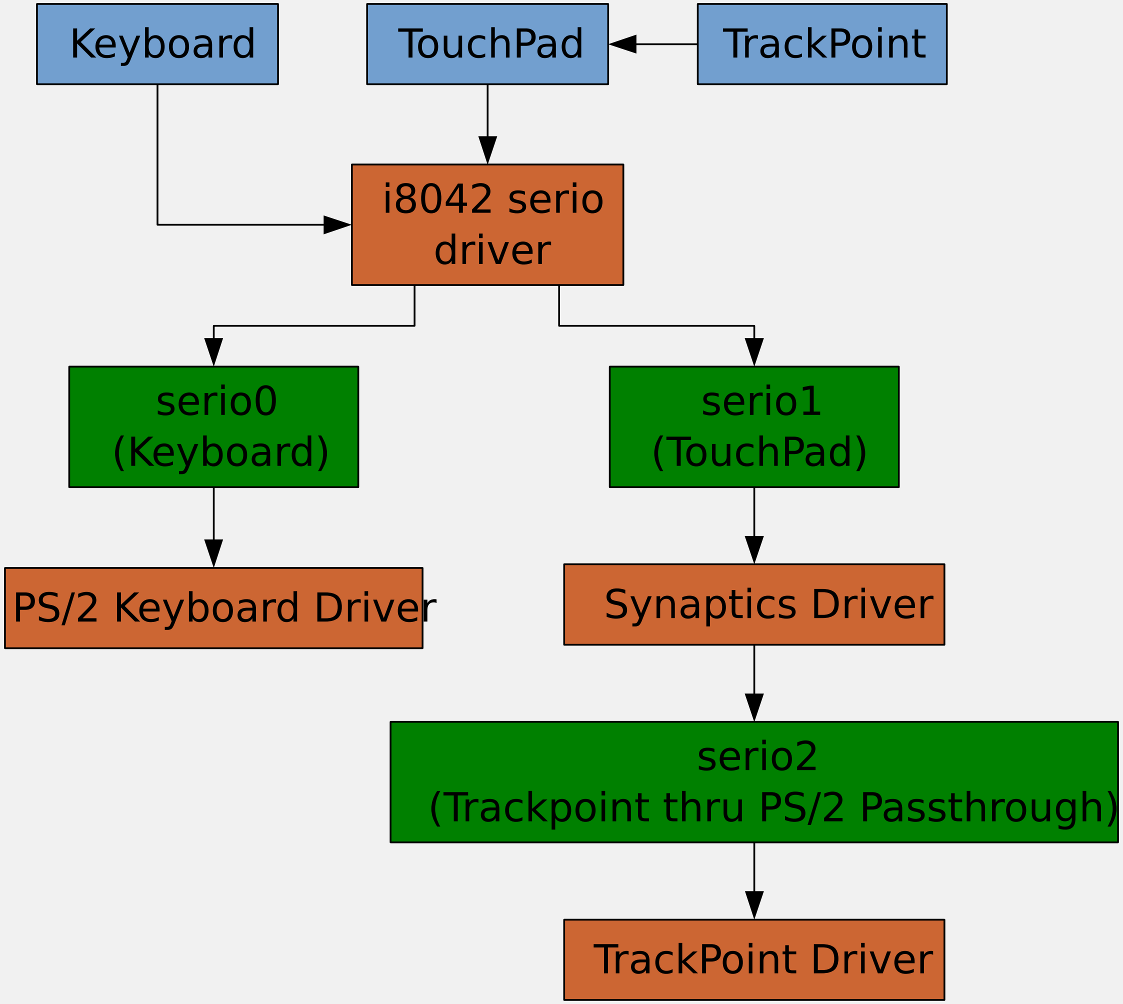
# Why is it still used?

- Designed when PS/2 was the only option
- Older versions of Windows still need it
- Mouse support in UEFI (PS/2 doesn't need special drivers)

# Alternatives

- I$^2$C: Used by Chromebooks, some Dells
- RMI4/SMBus: Supported by Lenovo ThinkPads and some HP notebooks. Linux support is a WIP.
- Not widely used, but some laptops have them

# PS/2 in the kernel

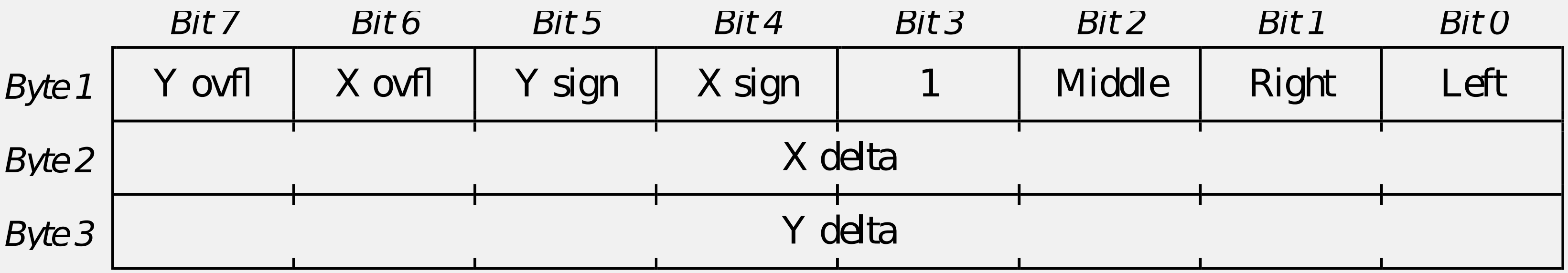


# The basic PS/2 protocol for mice

- Each packet is 11 bits
- Payload is 8 bits
- Packets are sent in groups of 3 for normal 3 button mice, groups of 4 for mice with 5 buttons
- There are some commands you can send:
  - 0xFF – Reset the mouse
  - 0xFE – Resend
  - 0xF6 – Set defaults
  - Etc.

Diagram from Synaptics PS/2 TouchPad Interfacing Guide

# The basic PS/2 protocol for mice



| | *Bit 7* | *Bit 6* | *Bit 5* | *Bit 4* | *Bit 3* | *Bit 2* | *Bit 1* | *Bit 0* |
|---|---|---|---|---|---|---|---|---|
| *Byte 1* | Y ovfl | X ovfl | Y sign | X sign | 1 | Middle | Right | Left |
| *Byte 2* | X delta | | | | | | | |
| *Byte 3* | Y delta | | | | | | | |

*Figure 3-2. PS/2 relative motion packet*

Diagram from Synaptics PS/2 TouchPad Interfacing Guide

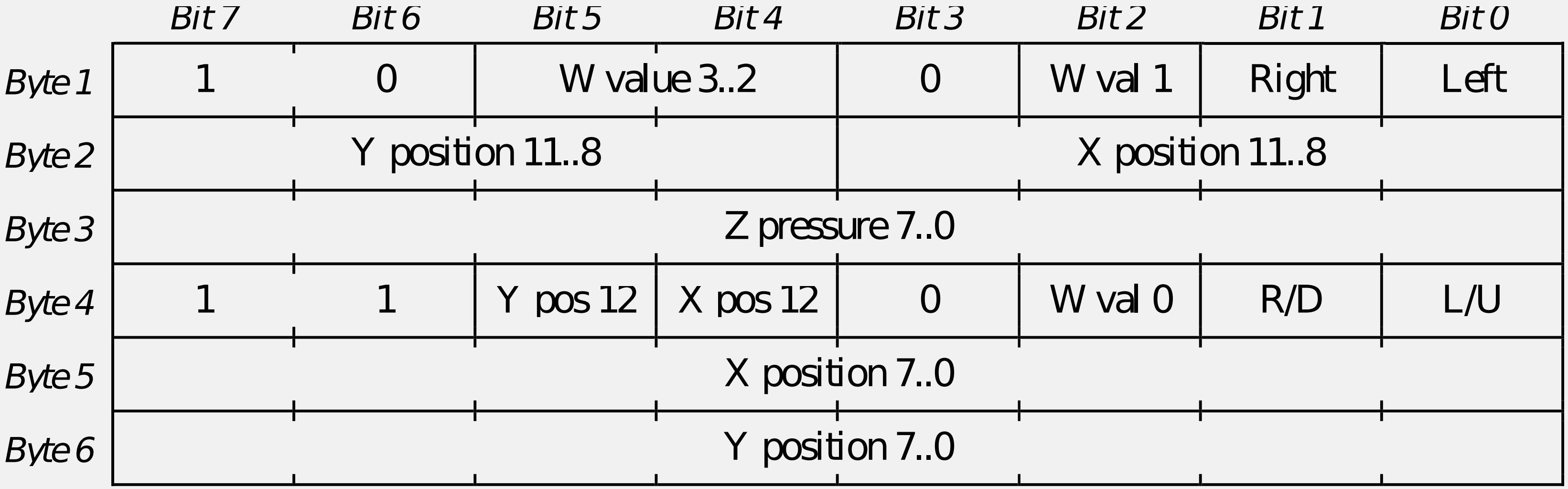
# Problems with PS/2

- i8042 controllers are picky, reject anything that isn't PS/2
- Some even modify packets
- Can't do absolute positioning, pressure, etc.

# Working around PS/2

- PS/2 alone wasn't sufficient for touchpads
- Manufacturers made their own protocols over PS/2
- Add more information into PS/2 packets
- Looks like normal PS/2 events to the i8042

# Example of a packet from SynPS/2



| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Byte 1 | 1 | 0 | W value 3..2 | | 0 | W val 1 | Right | Left |
| Byte 2 | Y position 11..8 | | | | X position 11..8 | | | |
| Byte 3 | Z pressure 7..0 | | | | | | | |
| Byte 4 | 1 | 1 | Y pos 12 | X pos 12 | 0 | W val 0 | R/D | L/U |
| Byte 5 | X position 7..0 | | | | | | | |
| Byte 6 | Y position 7..0 | | | | | | | |

*Figure 3-4. PS/2 absolute X/Y/Z/W motion packet (Wmode = 1)*

Diagram from Synaptics PS/2 TouchPad Interfacing Guide

# Working around PS/2

- Touchpads don't use extended protocols by default, so they work without special drivers
- Activated with special command sequences
- For Synaptics: 0xE8 (set sample rate) with the two bit argument containing part of the special command

# Working around PS/2

- For some touchpads, PS/2 is a secondary protocol (e.g. RMI4)
- Doesn't always get the same QA testing
- Quirks for these touchpads are very common
- Quirks both in the protocol and the touchpad

# ps2emu

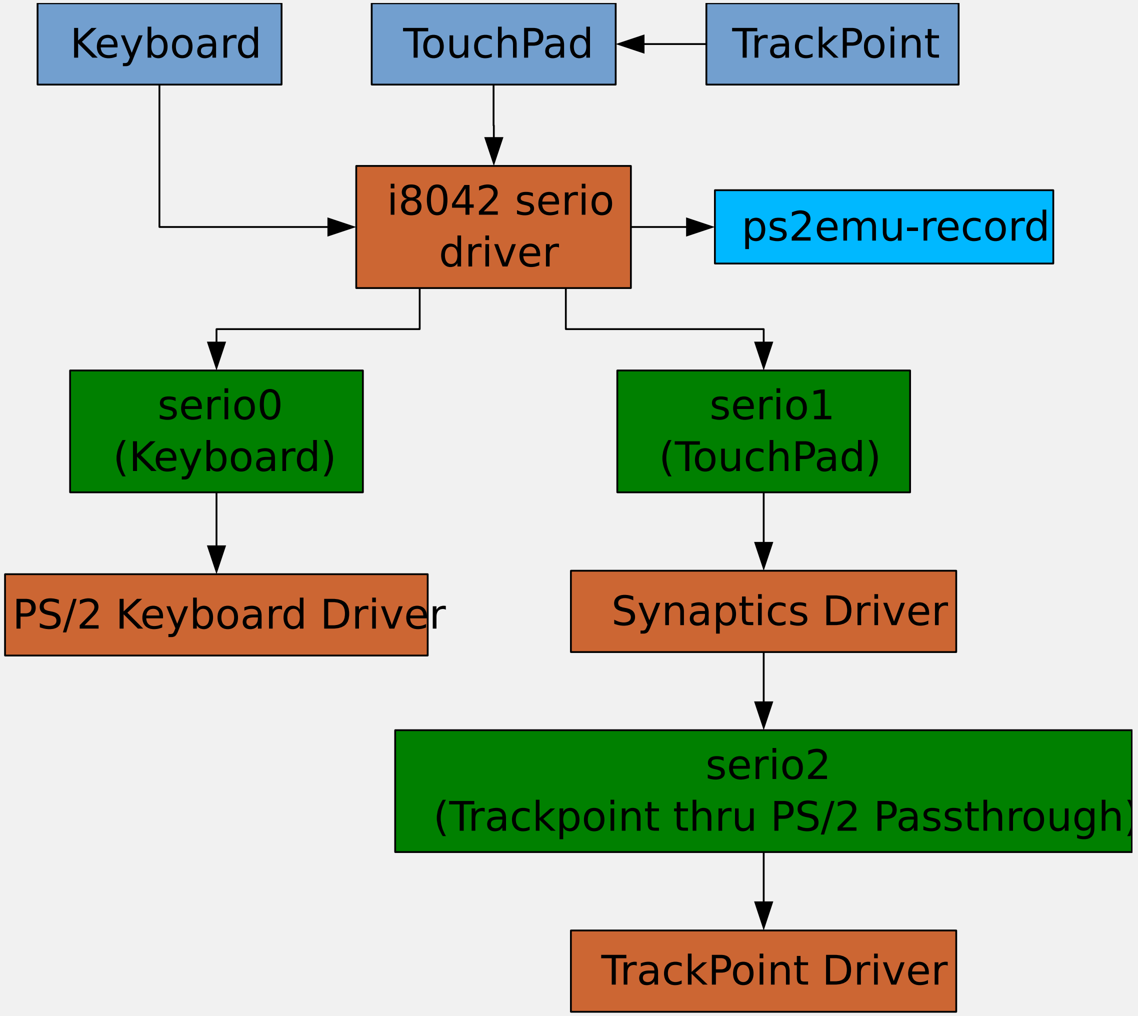
# ps2emu

- A new thing I made!
- My intern project at Red Hat
- Allows us to record PS/2 devices
- Recordings can be replayed on other machines
- Lets us reproduce bugs in PS/2 drivers locally
- Has helped fix various bugs in touchpad drivers

# ps2emu

- Two commands:
  - `ps2emu-record`: Records the PS/2 devices
  - `ps2emu-replay`: Replays the PS/2 devices
- One kernel module:
  - `userio`: Allows for userspace to create virtual serio ports to communicate to the kernel with

# ps2emu-record



# ps2emu File Format

```
T: A
S: Init
E: 0          S f2 # (parameter)
E: 2725       R fa # (interrupt, 1, 12)
E: 3900       R 00 # (interrupt, 1, 12)
E: 4356       S f2 # (parameter)
E: 7038       R fa # (interrupt, 1, 12)
E: 8265       R 00 # (interrupt, 1, 12)
E: 8479       S f6 # (parameter)
E: 11181      R fa # (interrupt, 1, 12)
E: 11378      S f3 # (parameter)
E: 14214      R fa # (interrupt, 1, 12)
…
```

Demonstration!

# Limitations of ps2emu

- Doesn't emulate a touchpad, just repeats whatever is in the log
- Interaction between driver and ps2emu must be the same as the log
  - Recordings may not work between kernel versions

# Regression tests?

- No working solution yet
- ps2emu-replay isn't enough for this, we need to be smarter
- Must not break whenever the driver changes

# Regression tests?

- Mapping TouchPad registers:
  - Not all TouchPads expose registers
  - Complex
  - Need every register value or it won't work

# Regression tests?

- Emulating a TouchPad
  - Very complex
  - Dummy device might be prone to bugs
  - Need one recording for each TouchPad

# Summary

- PS/2
    - Most common protocol in use for touchpads
    - Old, very backwards compatible
    - Transparent to the i8042
    - Manufacturer specific protocols
- ps2emu
    - Records and replays PS/2 devices
    - Potential for regression tests

# Links

1) Diagrams of packet layouts taken from *Synaptics PS/2 TouchPad Interfacing Guide* http://www.synaptics.com/sites/default/files/511-000275-01_RevB.pdf

- ps2emu userspace tools available here:
  - Git: https://github.com/Lyude/ps2emu
  - Fedora copr: https://copr.fedoraproject.org/coprs/lyude/ps2emu-tools/
- userio available here:
  - Git: https://github.com/Lyude/ps2emu-kmod

# Summary

- PS/2
  - Most common protocol in use for touchpads
  - Old, very backwards compatible
  - Transparent to the i8042
  - Manufacturer specific protocols
- ps2emu
  - Records and replays PS/2 devices
  - Potential for regression tests

# Summary

- PS/2
  - Most common protocol in use for touchpads
  - Old, very backwards compatible
  - Transparent to the i8042
  - Manufacturer specific protocols
- ps2emu
  - Records and replays PS/2 devices
  - Potential for regression tests