The userio Protocol

Copyright: © 2015 Stephen Chandler Paul <thatslyude@gmail.com>

Sponsored by Red Hat

Introduction

This module is intended to try to make the lives of input driver developers easier by allowing them to test various serio devices (mainly the various touchpads found on laptops) without having to have the physical device in front of them. userio accomplishes this by allowing any privileged userspace program to directly interact with the kernel's serio driver and control a virtual serio port from there.

Usage overview

In order to interact with the userio kernel module, one simply opens the /dev/userio character device in their applications. Commands are sent to the kernel module by writing to the device, and any data received from the serio driver is read as-is from the /dev/userio device. All of the structures and macros you need to interact with the device are defined in <linux/userio.h> and <linux/serio.h>.

Command Structure

The struct used for sending commands to /dev/userio is as follows:

```
struct userio_cmd {
        __u8 type;
        __u8 data;
};
```

`type` describes the type of command that is being sent. This can be any one of the USERIO_CMD macros defined in <linux/userio.h>. `data` is the argument that goes along with the command. In the event that the command doesn't have an argument, this field can be left untouched and will be ignored by the kernel. Each command should be sent by writing the struct directly to the character device. In the event that the command you send is invalid, an error will be returned by the character device and a more descriptive error will be printed to the kernel log. Only one command can be sent at a time, any additional data written to the character device after the initial command will be ignored.

To close the virtual serio port, just close /dev/userio.

Commands

USERIO_CMD_REGISTER

Registers the port with the serio driver and begins transmitting data back and forth. Registration can only be performed once a port type is set with USERIO_CMD_SET_PORT_TYPE. Has no argument.

USERIO_CMD_SET_PORT_TYPE

Sets the type of port we're emulating, where `data` is the port type being set. Can be any of the macros from <linux/serio.h>. For example: SERIO_8042 would set the port type to be a normal PS/2 port.

USERIO_CMD_SEND_INTERRUPT

Sends an interrupt through the virtual serio port to the serio driver, where `data` is the interrupt data being sent.

Userspace tools

The userio userspace tools are able to record PS/2 devices using some of the debugging information from i8042, and play back the devices on /dev/userio. The latest version of these tools can be found at:

> https://github.com/Lyude/ps2emu