Disabling the Webcam or USB Ports on Linux

karlcode.owtelse.com/blog/2017/01/09/disabling-usb-ports-on-linux

By Karl Roberts

Why disable a USB port?

You may just want to make sure that you webcam, usually connected to the internal USB bus, is turned off.

In my case my laptop webcam has become intermittently faulty. This has caused my laptop to hang on shutdown while the kernel tries to power down the device.

Clearly I want to disable the camera's port without disabling all the other USB ports on the laptop…

Identify the port

The first step was to identify the camera's port.

You can trawl through `dmesg` logs looking at all USB entries until you find the webcam, e.g.

```
$ dmesg
...
[    2.996705] usb 1-1.6: Product: Webcam SC-13HDL11624N
...
```

Here you can see that my webcam was on bus 1 - port 1 on the 6th device on that port e.g. `1-1.6`

However Linux has some commands and a pseudo "sys" file system that make it easier, for instance you could use `lsusb`

```
$ lsusb -t
/:  Bus 03.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/2p, 5000M
/:  Bus 02.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/2p, 480M
    |__ Port 1: Dev 2, If 0, Class=Mass Storage, Driver=usb-storage, 480M
    |__ Port 2: Dev 3, If 0, Class=Human Interface Device, Driver=usbhid, 12M
    |__ Port 2: Dev 3, If 1, Class=Human Interface Device, Driver=usbhid, 12M
/:  Bus 01.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/3p, 480M
    |__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/6p, 480M
        |__ Port 2: Dev 3, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M
        |__ Port 5: Dev 4, If 0, Class=Wireless, Driver=btusb, 12M
        |__ Port 5: Dev 4, If 1, Class=Wireless, Driver=btusb, 12M
        |__ Port 6: Dev 5, If 0, Class=Video, Driver=uvcvideo, 480M
        |__ Port 6: Dev 5, If 1, Class=Video, Driver=uvcvideo, 480M
```

If we walk this tree we can see bus 01 Port 1 is a USB hub and port 6 on the hub is a video device, which is my webcam... but this is a bit obtuse and confusing, and it doesn't yield the literal string `1-1.6` that I'll need to disable the webcam port.

A better method is to walk the `sys` pseudo file system that gives info on all devices attached to the kernel system.

```
$ for device in $(ls /sys/bus/usb/devices/*/product); do echo $device;cat $device;done
/sys/bus/usb/devices/1-1.2/product
Dell USB Entry Keyboard
/sys/bus/usb/devices/1-1.6/product
Webcam SC-13HDL11624N
/sys/bus/usb/devices/2-1/product
Amazon Kindle
/sys/bus/usb/devices/2-2/product
USB Receiver
/sys/bus/usb/devices/usb1/product
EHCI Host Controller
/sys/bus/usb/devices/usb2/product
xHCI Host Controller
/sys/bus/usb/devices/usb3/product
xHCI Host Controller
```

Here we can see that the file `/sys/bus/usb/devices/1-1.6/product` contains the `Webcam SC-13HDL11624N`

So in this case the USB device (or port) we need is `1-1.6` .

Because the Webcam is attached to the internal USB hub, it will always be listed at the same address, this is helpful as it means I can hard-code `1-1.6` where I need it rather than parsing the output of the command.

Turn off power to the Webcam

Now we know the USB device number, it is simple to turn off the camera using the `sys` file system.

By writing values to the "files" in the `sys` file system you can effect the devices that the file represents. Obviously you need to be root to do this, or be a user that has `sudo` permissions.

On Ubuntu

We can send a command to the USB driver to unbind a port,

```
$ echo '1-1.6' | sudo tee /sys/bus/usb/drivers/usb/unbind
```

Obviously replace "1-1.6" with whichever usb port your webcam is on (see above).

To turn it back on,

```
$ echo '1-1.6' | sudo tee /sys/bus/usb/drivers/usb/bind
```

This command pushed the USB device name, "1-1.6", to a special socket that acts like a command API to the USB driver, in other words rather than directly control the power of the device we ask the USB driver to do it for us.

Run it at start-up

Now I have control over my USB ports I want to disable this port at start-up. Ubuntu uses the `anacron` cron daemon which allows a special syntax, `@reboot`, to hook a command to the reboot sequence.

Simple edit root's crontab

```
$ sudo crontab -e
```

And then append the following

```
@reboot echo '1-1.6' > /sys/bus/usb/drivers/usb/unbind
```

Again edit '1-1.6' to your usb device's number.

And that's it. My laptop can now shutdown without hanging.

Cheers.