

List of Slides

– General Linux 2 –

Set Up USB devices [2]

(Linux Professional Institute Certification)

a

```
.~.  
/V\   by: geoffrey robertson  
//  \\\   geoffrey@zip.com.au  
@._. @
```

\$Id: gl2.101.5.slides.tex,v 1.3 2003/08/29 14:36:14 waratah Exp \$

^aCopyright © 2002 Geoffrey Robertson. Permission is granted to make and distribute verbatim copies or modified versions of this document provided that this copyright notice and this permission notice are preserved on all copies under the terms of the GNU General Public License as published by the Free Software Foundation—either version 2 of the License or (at your option) any later version.

Set Up USB devices [2]

(1.1) 101 Hardware & Architecture [11]

2.101.1 Configure BIOS Fundamental Settings [3]

2.101.2 Setup SCSI devices [2]

2.101.3 Setup different PC expansion cards [2]

2.101.4 Configure communication devices [2]

2.101.5 Set Up USB devices [2]

Set Up USB devices [2]

Objective

Candidates should be able to activate USB support, use and configure different USB devices. This objective includes the correct selection of the USB chipset and the corresponding module. It also includes the knowledge of the basic architecture of the layer model of USB as well as the different modules used in the different layers.

Set Up USB devices [2]

Key files, terms, and utilities

```
lspci(8)  
usb-uhci.o  
usb-ohci.o  
/etc/usbmgr/  
usbmodules  
/etc/hotplug
```

Set Up USB devices [2]

Resources of interest

The Linux-USB Project :

`http://www.linux-usb.org`:

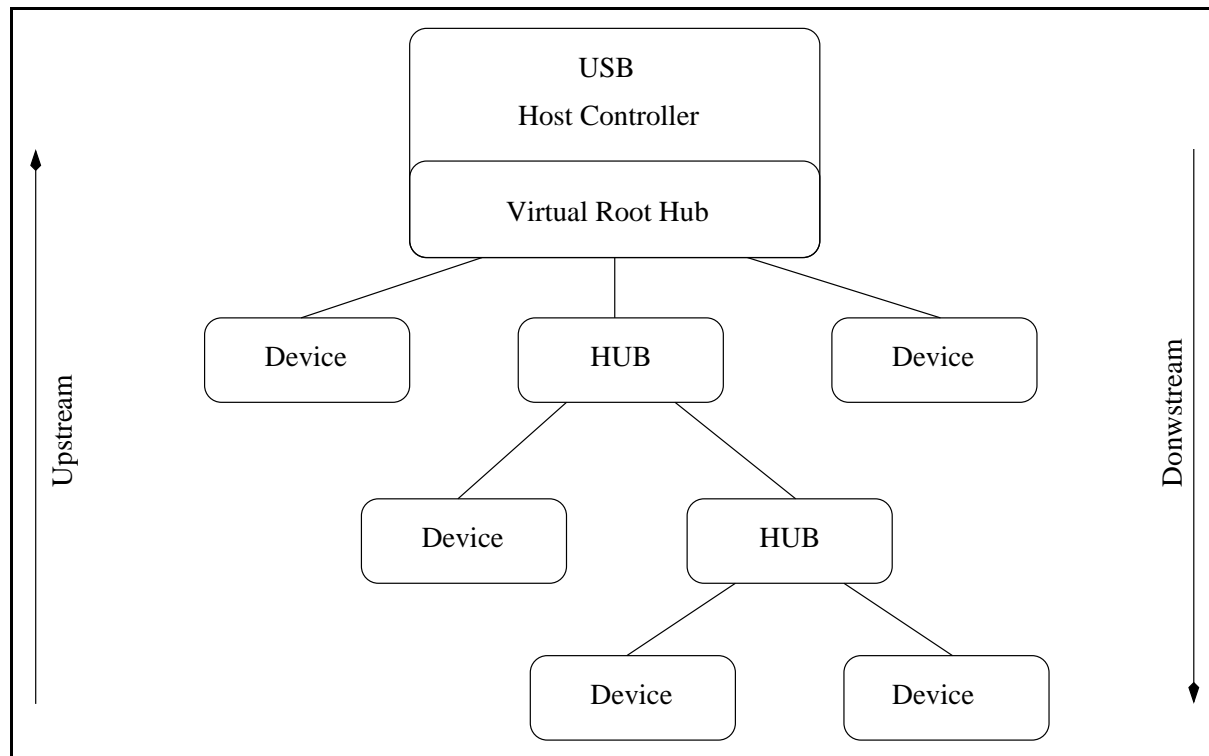
The Linux USB Sub System : by Brad Hards, Sigma Bravo Pty Ltd

The Universal Serial Bus

- A serial transmission scheme
- Two versions of USB Version 1 & Version 2
- Version 1
 - released January 1996
 - supports speeds up to 12MBit/s (8.5Mbit/s in practice)
 - supports up to 127 devices connected to the bus
- Version 2:
 - announced 1999
 - supports speeds up to 480Mbit/s
- Devices can be self or bus powered

USB Topology

The system unit contains the host controller and one virtual root hub with at least one (and normally two) USB interfaces. These interfaces can then be connected directly to a USB device or to another HUB.

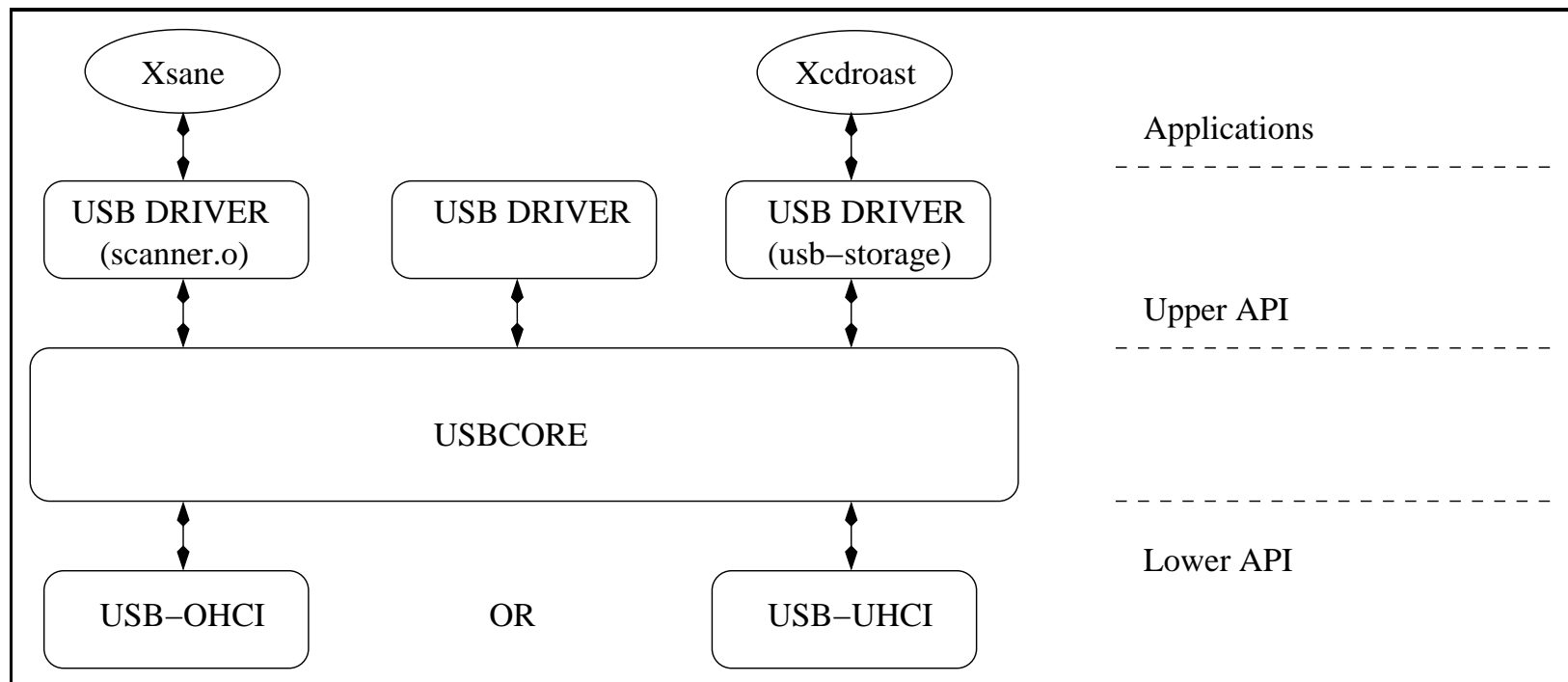


USB Device Driver Layers

The Device drivers for the USB sub-system are split into two distinct layers:

Hardware Layer usbcore & usb-uhci / usb-ohci

API Layer - Application / Product specific



USB Controllers

There are two categories of USB controller

usb-uhci For Intel, PIIX4, Via controllers

usb-ohci For Compaq, iMacs, OPTi, SiS, ALi controllers

The UHCI controllers use a 16 bit IO address:

I/O at 0xHHHH eg: I/O at 0xe400

The OHCI controllers use a 32 bit memory address:

memory at 0xHH000000 eg memory at 0xee000000

USB Controllers

To determine your controller type, examine `/proc/pci` for a clue:

```
[root@Node4] root]# cat /proc/pci
```

```
PCI devices found:
```

```
.....
```

```
Bus 0, device 7, function 2:
```

```
USB Controller: VIA Technologies Inc. UHCI USB(rev 17).
```

```
IRQ 10.
```

```
Master Capable. Latency=32.
```

```
I/O at 0xe400 [0xe41f].
```

```
.....
```

USB Modules

Assuming you have a modular kernel, the following modules will be required:

usbcore The base usb kernel module

plus one of the controller specific modules either

usb-uhci For Intel, PIIX4, Via controllers

usb-ohci For Compaq, iMacs, OPTi, SiS, ALi controllers

USB Modules

Configuration

An entry in `/etc/modules.conf` aliases the specific controller to `usb-controller` as follows:

```
alias usb-controller usb-uhci
```

USB Modules

Starting up the USB sub-system

To have the usb sub-system startup automatically at boot time, all you need to do is ensure that the above alias line is present in `/etc/modules.conf`.

To startup manually, do the following steps:

- `insmod usbcore`
- `insmod usb-uhci` (or `usb-ohci`)
- mount the `usbdevfs` filesystem (optional but highly recommended)

USB Modules

Example

```
[root@Node4]# insmod usbcore
Using /lib/modules/2.4.18-4/kernel/drivers/usb/usbcore.o
```

```
[root@Node4] root]# insmod usb-uhci
Using /lib/modules/2.4.18-4/kernel/drivers/usb/usb-uhci.o
```

```
[root@Node4]# mount -t usbdevfs usbdevfs /proc/bus/usb
```

Once this is done, you should see the following entries in /proc/bus/usb:

```
[root@Node4] root]# ls /proc/bus/usb
001  devices  drivers
```

USB Interrogation Utilities

LSUSB - A console view of USB devices

Lsusb is a text utility contained in the usbutils package. Use 'rpm -Uvh usbutils.xxx.rpm' to install.

```
[root@node4]# lsusb
```

```
Bus 001 Device 001: ID 0000:0000 Virtual Hub
```

```
Device Descriptor:
```

```
  bLength                18
  bDescriptorType        1
  bcdUSB                  1.00
  bDeviceClass            9  Hub
  iProduct                2  USB UHCI Root Hub
```

```
.....
```


LSUSB - A console view of USB devices

Bus 001 Device 002: ID 03f0:0601 Hewlett-Packard ScanJet 6300c

Device Descriptor:

bLength	18	
bDescriptorType	1	
bcdUSB	1.00	
bDeviceClass	0	Interface
bDeviceSubClass	0	
bDeviceProtocol	0	
bMaxPacketSize0	8	
idVendor	0x03f0	Hewlett-Packard
idProduct	0x0601	ScanJet 6300c
bcdDevice	1.00	
iManufacturer	1	
iProduct	2	HP ScanJet 6300C
iSerial	3	SG9941706SPE

LSUSB - A console view of USB devices

Bus 001 Device 003: ID 1189:6000

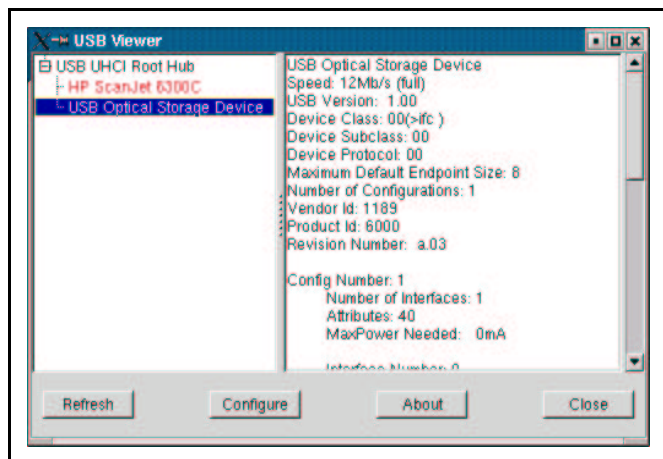
Device Descriptor:

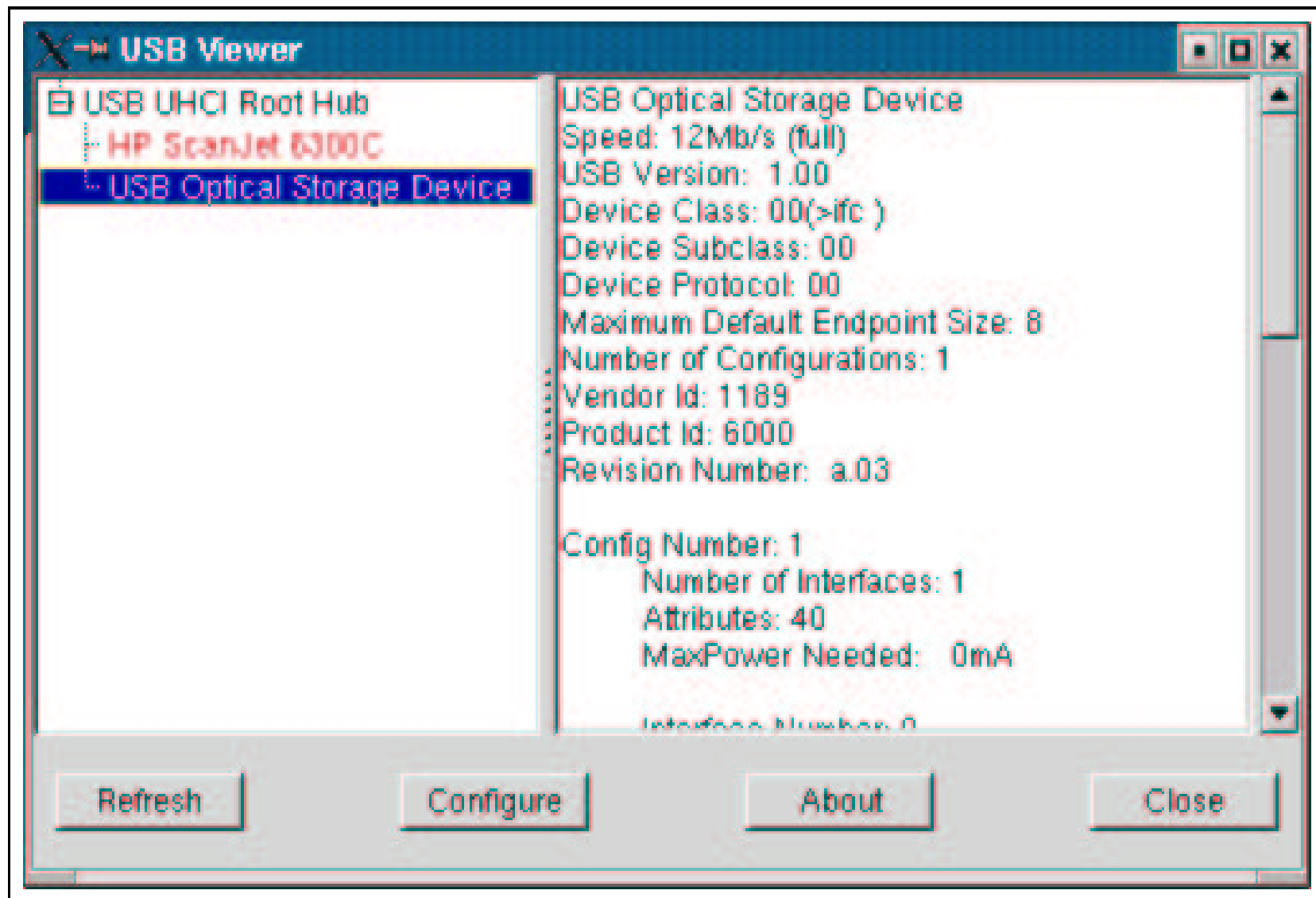
bLength	18	
bDescriptorType	1	
bcdUSB	1.00	
bDeviceClass	0	Interface
bDeviceSubClass	0	
bDeviceProtocol	0	
bMaxPacketSize0	8	
idVendor	0x1189	
idProduct	0x6000	
bcdDevice	a.03	
iManufacturer	0	
iProduct	1	USB Optical Storage Device
iSerial	0	

USB Interrogation Utilities

USBVIEW - An X view of USB devices

- Usbview is a GUI utility contained in the usbview package. Use 'rpm -Uvh usbview.rpm' to install.
- Usbview parses /proc/bus/usb/devices for connected USB devices. Any device that has a problem will be printed in red.





Hotplugging USB Devices

When a device is plugged into a USB port, it will automatically register itself with the USB subsystem. The upper API drivers will not however automatically 'insmod' themselves unless the hotplug package has been installed.

With the hotplug package installed, an entry in `/proc/sys/kernel/hotplug` will be created which will contain the name of an executable to be called whenever a new device is detected on the bus.

```
$ ls /proc/sys/kernel/hotplug  
/sbin/hotplug
```

Hotplugging USB Devices

For example, when a USB scanner is plugged in, hotplug will automatically load the module 'scanner.o'. The xsane application can then be run directly without any user intervention.

- `/sbin/hotplug` is an executable which is called by the kernel (kernel space to user space interface)
- `/etc/hotplug` is a directory containing configuration information for hotplug (which drivers to load when a device is plugged in)

The End