How to find Linux kernel driver associated to a device

Contents

How to find Linux kernel driver associated to a device

Stable: 18.09.2019 - 09:58 / Revision: 18.09.2019 - 09:56

Contents

1 Introduction

This article shows the user how to find the Linux® kernel driver associated to a kernel device.

This can, for example, be useful when debugging devices that the user does not know, or monitoring for correct system behavior.

2 Find kernel driver for a device

2.1 Major and minor numbers for a Linux kernel device

The device files in the Linux kernel are associated to a MAJOR and a MINOR number, giving each file a unitary identity. This abstraction of device handling is a basic features of the Linux kernel.

A list of MAJOR numbers, and rules for MINOR numbers are given in *Documentation/admin-guide/devices.txt* of the Linux kernel source[1], or in kernel.org[2].

2.2 List of available devices

A list of the available devices for the Linux kernel can be read from the procfs file */proc/devices*:

```
Board $> cat /proc/devices
```

This lists all of the available devices, according to their classification as a character or a block device.
The number preceding the device name corresponds to the MAJOR number of the device (for example, "4" is the MAJOR number for the "tty" device):

```
Character devices:
  1 mem
  2 pty
  3 ttyp
  4 /dev/vc/0
  4 tty
  5 /dev/tty
  5 /dev/console
  5 /dev/ptmx
  5 ttyRPMSG
  7 vcs
 10 misc
 13 input
 21 sg
 29 fb
 81 video4linux
 89 i2c
 90 mtd
116 alsa
128 ptm
136 pts
153 spi
166 ttyACM
180 usb
189 usb_device
199 galcore
226 drm
245 cec
246 media
247 ttySTM
248 bsg
249 watchdog
250 iio
251 ptp
252 pps
253 rtc
254 gpiochip

Block devices:
  1 ramdisk
  7 loop
  8 sd
 11 sr
 31 mtdblock
 65 sd
 66 sd
 67 sd
 68 sd
 69 sd
 70 sd
 71 sd
128 sd
129 sd
130 sd
131 sd
132 sd
```

```
133 sd
134 sd
135 sd
179 mmc
254 virtblk
259 blkext
```

For further information about the major and minor numbers for a Linux kernel driver, refer to the Linux tutorial web page[3].

Note: 'Misc devices have a specific setup; you can find the list of misc devices with the corresponding MINOR number in the */proc/misc* file.

2.3 Device entries in /dev

Each device has a corresponding entry in the /dev directory of the Linux kernel pseudo filesystem.

```
Board $> ls -lR /dev
```

Be careful, /dev contains some sub-directories containing device entries, that is, *input*. That the reason why *-R* should be used.

This command lists all of the device entries, including the device type and the associated MAJOR and MINOR numbers

For example:

```
crw-rw---- 1 root video 81, 0 Dec 18 16:26 video0
```

This device *video0* is of type character (c), with MAJOR number of 81 and MINOR number of 0.

2.4 System device entries in /sys/dev

All devices, classified by type (char or block), and identified by their MAJOR/MINOR number can be found in the *dev* subdirectory of the sysfs file system entry (*/sys*).

A platform device is then linked to each MAJOR/MINOR number.

For example:

```
Board $> ls -l /sys/dev/char/81\:0
lrwxrwxrwx 1 root root 0 Dec 18 17:00 81:0 -> ../../devices/platform/soc/4c006000.dcmi
/video4linux/video0
```

The device *video0* is linked to the platform device *4c006000.dcmi/video4linux/video0*.

2.5 Driver associated to a platform device

If the device is linked to a platform device, you can find the corresponding driver definition in the device tree with the compatible parameter.

For example: Look for device **4c006000.dcmi/video4linux/video0** in arch/arm/boot/dts/stm32mp157c.dtsi .

```
...
dcmi: dcmi@4c006000 {
 compatible = "st,stm32-dcmi";
 reg = <0x4c006000 0x400>;
 interrupts = <GIC_SPI 78 IRQ_TYPE_NONE>;
 resets = <&rcc CAMITF_R>;
 clocks = <&rcc DCMI>;
 clock-names = "mclk";
 dmas = <&dmamux1 75 0x400 0x05>;
 dma-names = "tx";
 status = "disabled";
};
...
```

The driver associated to the video0 device is ***st,stm32-dcmi***.

If the driver belongs to your Linux kernel tree, you can search for the driver by declaring st,stm32-dcmi as a compatible device.

- In the previous example, when looking for the driver compatible with *st,stm32-dcmi*, you find *drivers/media/platform /stm32/stm32-dcmi.c* driver

```
PC $> cd <your_kernel_source_path>
PC $> grep -rs "st,stm32-dcmi" *
...
drivers/media/platform/stm32/stm32-dcmi.c:        { .compatible = "st,stm32-dcmi"},
...
```

If the driver is not part of your Linux kernel source tree, it is present as a kernel object library file and you can check on the board:

```
Board $> cd /lib/modules/<kernel_version>
Board $> grep <compatible_name> modules.alias
```

This gives you the name of the module driver.

For example, for the gcnano driver used for the GPU:

```
Board $> grep "st,gcnano" modules.alias
 alias of:N*T*Cst,gcnano galcore
```

This means that the module name is ***galcore.ko***.

3 References

- Documentation/admin-guide/devices.txt
- https://www.kernel.org/doc/Documentation/admin-guide/devices.txt
- http://www.linux-tutorial.info/modules.php?name=MContent&pageid=94

Process File System (See https://en.wikipedia.org/wiki/Procfs for more details)

System File System (See https://en.wikipedia.org/wiki/Sysfs for more details)

Generic Interrupt Controller

Serial Peripheral Interface

Digital Camera Memory Interface

Graphics Processing Units

Permission error

Stable: 31.01.2020 - 13:23 / Revision: 31.01.2020 - 13:16

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST_editors, ST_readers, Selected_editors, sysop, reviewer