Kernel Internship Report (Outreachy)

Julia Lawall (Inria/LIP6)

Georgiana Chelu    Meghana Madhyastha    Aishwarya Pant

https://www.outreachy.org

October 23, 2018
What is Outreachy?

- Organized by the Software Freedom Conservancy
  - Formerly OPW, organized by Gnome.

- Goal: Get more women and other underrepresented groups into open source.

- Internship:
  - 3 months
  - $5,500 stipend
  - Paired with mentor

- Timing: May – August, December – March.
Who can apply for an internship?

• Women (cis and trans), trans men, and genderqueer people.

• Additionally, Outreachy is open to residents and nationals of the United States of any gender who are Black/African American, Hispanic/Latin@, American Indian, Alaska Native, Native Hawaiian, or Pacific Islander.

• Anyone who faces systematic bias or discrimination in the technology industry of their country is invited to apply.

• Must be able to work full time.

• Work is done remotely.

• Don’t have to be a student.
Application process

• 6 weeks, starting in mid February or early September.
• Starts with an eligibility check.
• Applicants work through the kernel tutorial and clean up staging drivers
  – https://kernelnewbies.org/Outreachyfirstpatch
• Applicants also work on project-specific tasks.
• Applicants complete an application, and are selected for the available projects.
Which projects are involved?

- Recent proposed kernel projects:
  - Summer 2018 (Round 16): Migrate NAND driver to new exec. op framework, Netfilter/Nftables, Non-cooperative userfaultfd stress test, GPU subsystem.

- Other projects:
  - Debian, GNOME, Mozilla, OpenStack, Wikimedia, etc.

Internships are financed by the project’s organization or by industry sponsors.
Contributions from applicants and interns

- 7 active applicants for 4 slots in Dec 2016-Mar 2017
  - 112 patches accepted during the application period (∼ 6 weeks)

- 6 active applicants for 2 slots in May-Aug 2018
  - 82 patches accepted during the application period
How can I help?

- Companies and individuals can:
  - Donate funds to support interns
  - Contact: organizers@outreachy.org

- Kernel developers can:
  - Review patches
  - Volunteer as mentors
  - Contact: Vaishali Thakkar <vthakkar1994@gmail.com> and Shraddha Barke <shraddha.6596@gmail.com>
Presentations from recent interns

- Georgiana Chelu: ADS1118 driver
- Meghana Madhyastha: Refactoring backlight and spi helpers in drm/tinydrm
- Aishwarya Pant: Improving attribute documentation

vthakkar1994@gmail.com, shraddha.6596@gmail.com
https://www.outreachy.org
organizers@outreachy.org
ADS1118 IIO Driver

Georgiana Chelu - Outreachy Intern

Daniel Baluta - Mentor
Who am I?

Graduated Computer Science in 2016

Graduated Network Security Master in 2018

Outreachy Intern Round 15 (December 2017 to March 2018)
ADS1118 was the start

- Analog-to-Digital Converter
- Temperature sensor
- Serial Peripheral Interface (SPI)
Its big brother ADS1015

- Analog-to-Digital Converter
- Inter-Integrated Circuit (I²C)
- The implementation is already in the Linux Kernel
Challenges

- The two ADC are very similar but have some “small” differences
- SPI vs I2C busses
  - use regmap to abstract this difference
- ADS1015 has 4 registers
- ADS1118 has only two registers (doesn’t have support for events)
- hardware arrived late
- SPI reads didn’t work at the beginning
  - we used a digital signal analyzer to debug the problems
Development Setup

Diolan + Logic Analyzer + ADS1118 = Development Setup
SPI interface

- there are four important logic signals:
  - MOSI (master-out-slave-in)
  - MISO (master-in-slave-out)
  - CS  (chip-select)
  - SCLK (serial clock)

- use an analyzer to better understand the SPI protocol
Driver development

Hardware:

- ADS1118 and ADS1015 have a lot of similarities
- the main difference is the communication interface

Solution:

- Split the exiting driver into ti-ads1015.c and ti-ads1015_i2c.c
- Add the SPI communication to ti-ads1015_spi.c
- relatively simple because of regmap API
● i2c + core split done, v1 sent to Linux IIO list
● v2 is ready to be reviewed but we want to integrated SPI changes too
● spi part almost done
  ○ bugs after doing second read
Summary

- Many thanks to my mentor Daniel Baluta
- A lot of thanks to Julia Lawall, Greg KH and Jonathan Cameron
- The project was a challenge for me, but a fun experience
- The analyzer is the best tool
- I learnt a lot about IIO Framework
Thank you!
Refactoring backlight and spi helpers in drm/tinydrm

Meghana Madhyastha
Outline

• About me
• Introduction
  – Project Goals
  – DRM
  – TinyDRM
• Backlight
• SPI
• Conclusion
About Me

• Round 15 (Dec 2017-Feb 2018) Outreachy intern
• Mentored by Daniel Vetter, Sean Paul and Noralf Trønnes to contribute to the drm subsystem.
• Currently a first year PhD student at Johns Hopkins University working on scalable machine learning
Project Goals

• Refactor Backlight and SPI helpers in drm/tinydrm
• Make the helpers less tinydrm specific and make them generic so that they can be used by other drivers
Introduction: DRM

• Direct Rendering Manager
• Subsystem of the linux kernel
• Exposes an API that user space programs can use to send commands and data to the GPU.
• Addresses limitation of fbdev: able to handle modern 3D accelerated GPU based video hardware
Introduction: DRM
Introduction: Tinydrm

- Driver helpers for very simple display hardware.
- `drm_simple_display_pipe` coupled with a `drm_connector` which has only one fixed `drm_display_mode`.
- Framebuffers - cma helper
- Support framebuffer flushing (dirty).
- Support fbdev
Introduction: Tinydrm
Introduction: Tinydrm

Task: Refactor and move helpers from tinydrm-helpers to general drm source code files so that they can be used by other drivers.
Backlight

- Previously: Helpers present in tinydrm to find, enable and disable backlight
- The task: Backlight is used by other drivers in drm. Can we make the helpers general? Can we move them to video/backlight?
THEN

- tinydrm/helpers
  - Usage:
    if (ddata->backlight) {
      ddata->backlight->props.power = FB_BLANK_UNBLANK;
      backlight_update_status(ddata->backlight);
    }

NOW

- video/backlight/backlight.c
  Separate function for enabling and disabling backlight
  - static inline int backlight_enable(struct backlight_device *bd)
  - static inline int backlight_disable(struct backlight_device *bd)
  - Usage: backlight_enable(ddata->backlight);
THEN

- tinydrm/helpers
  - struct backlight_device
    *tinydrm_of_find_backlight(struct device *dev)
  - Usage: mipi->backlight = tinydrm_of_find_backlight(dev);

NOW

- video/backlight/backlight.c
  - struct backlight_device
    *of_find_backlight(struct device *dev)
  - Usage: mipi->backlight = of_find_backlight(dev);
• SPI: Interface bus - send data between microcontrollers and small peripherals (eg. shift registers, sensors, and SD cards.
• In Tinydrm: Helpers for device drivers to communicate with spi.
SPI Core:
• Checks if it can do DMA mapping.
• split a buffer into max_dma_len chunks for the spi controller driver to handle.
• The problem: Raspberry Pi and drivers/spi/spi-bcm2835.c - has an upper bound check on dma transfer length in bcm2835_spi_can_dma().
• Upper bound check happens before the tinydrm-spi helper splits the buffer into chunks
The solution

- Remove chunk splitting in tinydrm_spi_transfer in tinydrm-helpers i.e. let the spi-core do the chunk splitting.
- The spi core will split a buffer into max_dma_len chunks for the spi controller driver to handle.
- Remove automatic byte swapping in tinydrm_spi_transfer as it doesn't have users.
- Remove the upper bound check on dma transfer length in bcm2835_spi_can_dma().
Conclusion

- Refactored backlight and spi helpers
- Learned a lot about the Linux kernel.
- Learned how to collaborate with people and communicate effectively.
Improving attribute documentation

Aishwarya Pant, Outreachy Intern
Mentor - Julia Lawall

@aishpant
whoami

• Backend Engineer at Monzo Bank 💰
• Outreachy intern with Linux Kernel from December 2017 – February 2018
• Interested in distributed systems
• Favourite emoji 👍
Linux ABI

- 🐧 **Linux Application Binary Interface** defines how userspace communicates with the kernel
  - mostly syscalls
  - pseudo file systems like procfs, debugfs, configfs, sysfs
  - iocls
Linux ABI

- No patch can break the userspace
- Unfortunately any such break is usually detected when userspace applications are already using it
- Documented behaviour can help users **other than developers** to test new interfaces
- 🤷‍♀️ Both testing and documentation efforts are essential for kernel development
Linux ABI - sysfs

• Virtual in-memory file system
• Exports information about internal kernel data structures and layout to userspace
• Do one thing well
  – attributes should export one value per file
  – values should be text-based and map to simple C types
Linux ABI - sysfs

- Mounted at `/sys`
- 2 years of backward compatibility is guaranteed for interfaces documented in *Documentation/ABI/stable*
• There are around 332 syscalls but about 30,000 sysfs files on my laptop
• **This is massive**, having only been introduced in kernel release version 2.5 with the unification of the device model
• 😞 Difficult to maintain
Improving sysfs documentation

• In a preliminary study of drivers, I found that ~1000 attributes are missing documentation
• sysfs documentation format -

- What: (the full sysfs path of the attribute)
- Date: (date of creation)
- KernelVersion: (kernel version it first showed up in)
- Contact: (primary contact)
- Description: (long description on usage)
abi2doc

- Helper tool that tries to generate and format sysfs style documentation
  - Prerequisite python3 and coccinelle

usage: abi2doc [-h] -f SOURCE_FILE -o OUTPUT_FILE

Helper for documenting Linux Kernel sysfs attributes

required arguments:
  -f SOURCE_FILE  linux source file to document
  -o OUTPUT_FILE  location of the generated sysfs ABI documentation

optional arguments:
  -h, --help      show this help message and exit
abi2doc

• Example usage -

```bash
abi2doc -f drivers/video/backlight/lp855x_bl.c -o sysfs_doc.txt
```

• **Date** and **KernelVersion** for found attributes are filled in accurately

• **Contact** details are prompted once per script run

• **What** and **Description** are prompted on every attribute
abi2doc example output

What: /sys/class/backlight/<backlight>/bled_mode
Date: Oct, 2012
KernelVersion: 3.7
Contact: dri-devel@lists.freedesktop.org
Description:

(W0) Write to the backlight mapping mode. The backlight current can be mapped for either exponential (value "0") or linear mapping modes (default).

%%%% Hints below %%%%%
bled_mode DEVICE_ATTR drivers/video/backlight/lm3639_bl.c 220

%%%% store fn comments %%%%%
/* backlight mapping mode */

%%%% commit message %%%%%
commit 8f59b588d11960ceafb42c4535dc73c2c5f3136c
Author: G.Shark Jeong <gshark.jeong@gmail.com>
Date: Thu Oct 4 17:12:55 2012 -0700

backlight: add new lm3639 backlight driver

This driver is a general version for LM3639 backlight + flash of TI.

LM3639:
The LM3639 is a single chip LCD Display Backlight driver +
camera driver. Programming is done over an I2C compatible
www.ti.com
abi2doc

- Added documentation for 312 sysfs attributes 🎉

![Graph showing the number of sysfs attributes documented over time](image)
What did I learn?

- 😞 Documentation within the kernel is kind of a mess
- 🖋️ Made me think about how to write and communicate effectively
- 📚 Overview of kernel subsystems
- 👤 More confident engineer and a better programmer because of the Outreachy experience
Fin

- Links
  - abi2doc source code
  - Documentation patches
  - Blog
  - @aishpant