

Getting Started With Linux Kernel Development

It's Just a Software Project

- Kernel development is not magic.
- Like any other large complex software system
 - Language
 - Domain knowledge
 - Development process

Awesome Benefits From Learning Linux Kernel Development

- Learning C will make you a better programmer.
- Coding C in the Linux environment will teach you about a lot of the software on your machine.
- Understand how your operating system works.
- Learn how hardware works, take the mystery out of devices.

Language: C

- Learn the language inside and out
 - **The C Programming Language**
Brian W. Kernighan and Dennis Ritchie
 - **The Linux Programming Interface**
Michael Kerrisk

Domain Knowledge

- OS theory and Linux kernel ‘theory’
 - **Operating System Concepts**
Abraham Silberschatz, Peter Baer Galvin, Greg Cagne
 - **Modern Operating Systems**
Andrew S. Tanenbaum
 - **Linux Kernel Development**
Robert Love
 - **Linux Device Drivers**
Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman

Development Process

- All in the open.
- All development and communication done via mailing list and patches (git).
- New contributors very welcome.

One Path

- Tool up.
- Lurk on the mailing lists.
- Checkpatch fixes to get a hang of the process.
- Do some refactoring (actually write some code).
- Build and test kernels and hardware.
- Write a device driver.

Tool Up

You're already building C code and using git, right?

- Set up your editor.
 - see `Documentation/process/coding-style.rst`
- Set up email
 - Retrieve, sort, view/edit, send
 - You will spend a lot of time in your email client
 - Normal communication (asking questions)
 - Reviewing (learning from) patches
 - Sending patches

Subscribe to Mailing Lists

- Kernel newbies mailing list
<https://kernelnewbies.org>
- Linux device driver mailing list
<http://linuxdriverproject.org>

Mailing List Tips

- Don't email a kernel developer without CC'ing a mailing list.
- Be polite.
- Research you question first.
- Be meticulous with spelling and grammar.

Set Up Kernel Tree

- Clone Greg Kroah-Hartman's tree

<https://git.kernel.org/pub/scm/linux/kernel/git/gregkh/staging.git/>

- Set up two tracking branches
 - staging-next
 - staging-testing

Scope

- `drivers/staging/*`
- `Documentation/process/*`
- `include/linux/*`
- `include/uapi/linux/*`

First Patch

- Run checkpatch.pl on drivers/staging/FOO

```
$ scripts/checkpatch.pl -f --terse --strict --show-types drivers/staging/FOO/*.c
```

- Pick one warning type to fix.
- Craft a single patch
see Documentation/process/submitting-patches.rst
- Submit patch to driver dev mailing list.
- For more in depth direction see
 - <https://kernelnewbies.org/FirstKernelPatch>
 - <http://tobin.cc/blog/kernel-dev-1>

Thanks for listening!

Tobin Harding

<http://tobin.cc>

me@tobin.cc