# Linux kernel Bug Fixing Mentorship Program Learning Resources

Author: Shuah Khan

This document is a compilation of a list of learning resources for mentees participating in the Linux kernel Bug Fixing Project.

Table of contents

## How and when to ask about a patch

Please don't send content free pings and please allow a reasonable time for review. People get busy, go on holiday, attend conferences etc., hence unless there is some reason for urgency (like critical bug fixes). Please allow at least a couple of weeks for review. If there have been review comments then people may be waiting for those to be addressed.

Sending content free pings adds to the mail volume (if they are seen at all) which is often the problem and since they can't be reviewed directly if something has gone wrong you'll have to resend the patches anyway, so sending again is generally a better approach though there are some other maintainers who like them - if in doubt look at how patches for the subsystem are normally handled.

## General Resources

- docs.kernel.org/Documentation/process/submitting-patches.rst
- https://lore.kernel.org
- https://kernel.org
- https://docs.kernel.org/
- Linux Kernel Development - Robert Love
- The Linux Programming Interface - Michael Kerrisk
  - https://archive.org/download/The_Linux_Programming_Interface/The_Linux_Programming_Interface.pdf
- Linux Device Drivers 3rd Edition
- Understanding the Linux Kernel
- Linux Insides
- **Checking features supported on a system**
  - get_feat.pl list
  - get_feat.pl list –arch=arm64
- Checking system calls supported on a system:
  - ausyscall –dump > syscalls_dump.out
- Discovering Linux kernel subsystems used by a workload
- Operating Systems: Three Easy Pieces
- The Linux Memory Manager
- The Linux Kernel Module Programming Guide
- File System Implementation
- itBooks/Linux Kernel Networking - Implementation and Theory.pdf at master

## Using  syzkaller

- Using syzkaller, part 1: Fuzzing the Linux kernel
  - https://www.col]labora.com/news-and-blog/blog/2020/03/26/syzkaller-fuzzing-the-kernel/
- Using syzkaller, part 2: Detecting programming bugs in the Linux kernel
  - Using syzkaller, part 2: Detecting programming bugs in the Linux kernel
- Using syzkaller, part 3: Fuzzing your changes
  - Using syzkaller, part 3: Fuzzing your changes
- Using syzkaller, part 4: Driver fuzzing
  - Using syzkaller, part 4: Driver fuzzing
- ▶ Finding Linux Kernel Bugs with Syzkaller: Debugging the Kernel pt2

## Setting up syz environment for reproducing and debugging problems

- syzkaller/setup.md at master - linux
- syzkaller/docs/linux/setup_ubuntu-host_qemu-vm_x86-64-kernel.md at master
- syzkaller/docs/linux/setup_linux-host_isolated.md at master

## Submitting patch to syzbot for testing

- Reference - syzbot.md - google/syzkaller · GitHub
- Does this problem you are debugging exist in Linus's master. Submit Linus's master for test to confirm:
  - #syz test: git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git master
- Does your patch fix the problem? Submit the patch for testing:
  - #syz fix <git repo> followed by patch
- Reference - https://groups.google.com/g/syzkaller-bugs/c/HW4z7PpD9B0/m/XQTdRRHrAwAJ
- Convert syz program to c: syz-prog2c - please note that this doesn't always work reliably.

## Event Tracing Resources

- Event Tracing — The Linux Kernel documentation
- An eBPF overview, part 5: Tracing user processes

## Getting started with Linux Kernel Debugging

- Debug kernel panics | Sanjeev Sharma Blog
- Understanding a Kernel Oops! - Open Source For You
- **Bug hunting — The Linux Kernel documentation**
- **Mentorship Session: Linux Kernel Debugging Tricks of the Trade**

## LF Live MentorshiLF Live: Mentorship Series | LF Eventsp Webinar Series

- **Dynamic Program Analysis for Fun and Profit: Mentorship Session: Dynamic Program Analysis for Fun and Profit | LF Events**
- **Fuzzing Linux Kernel: Mentorship Session: Fuzzing Linux Kernel | LF Events**

## Debugging techniques

- Tools and Techniques to Debug an Embedded Linux System - Sergio Prado, Embedded Labworks - Tools and Techniques to Debug an Embedded Linux System - Sergio Prado, Embedded Labworks
- Linux Kernel Debugging: Going Beyond Printk Messages - Sergio Prado, Embedded Labworks - https://www.youtube.com/watch?v=NDXYpR_m1CU
- Demystifying Linux Kernel Initialization - The New Stack
- This blog compares slub_debug, kasan, and kfence and summarizes their usage and limitations debugging mechanisms:
  - Linux SLUB Allocator Internals and Debugging - KFENCE, Part 4 of 4
- drgn
- https://elinux.org/Debugging_Portal
- Debugging The Linux Kernel Using Gdb - eLinux.org
- Using `gdb` to Debug the Linux Kernel — Star Lab Software
- Demystifying Linux Kernel Initialization

## Virtualization (qemu, kvm) resources

- GDB usage — QEMU documentation
- Buildroot
- Cloud Free Tier | Oracle India
- Prepare the environment for developing Linux kernel with qemu. | by DaeSeok Youn | Medium
- Setting up QEMU-KVM for kernel development
- Direct Linux Boot — QEMU documentation
- https://blog.cloudflare.com/the-tale-of-a-single-register-value/
- https://lwn.net/Articles/682540
- https://github.com/google/syzkaller/blob/master/docs/linux/setup_ubuntu-host_qemu-vm_x86-64-kernel.md
- Fixing bugs in the Linux kernel with Syzbot, Qemu and GDB – Javier Carrasco

## Finding bugs and KTODO resources

- Documentation/admin-guide/bug-hunting.rst
- Make Linux Developers Fix Your Kernel Bug
- How to Report and Handle Linux Kernel Regressions
- Run kselftest and kunit
- Monitor regressions.lists.linux.dev archive mirror
- kernel_debug_tricks.md · GitHub
- KTODO: add check for failure in function_something()
    - Then people can look on lore or use lei to find small tasks to work on or they could use lei.
    - lei q -I https://lore.kernel.org/all/ -o ~/Mail/KTODO --dedupe=mid 'KTODO AND rt:6.month.ago..'
    - Then grep ^KTODO ~/Mail/KTODO -R and cat the filename you want.

## Linux kernel subsystem resources

- Memory Management Resources
    - https://elinux.org/Memory_Management_Presentations
    - An Intro to the Linux Memory Access Workload Simulator (masim) - The New Stack
    - Device Driver primer: The Linux Kernel Module Programming Guide
    - Understanding The Linux Kernel, chapters 2, 8, 9
- devm_* api
    - docs.kernel.org/Documentation/driver-api/driver-model/devres.rst
    - docs.kernel.org/Documentation/driver-api/driver-model/design-patterns.rst
    - docs.kernel.org/Documentation/driver-api/device-io.rst
- Linux Internals : Interprocess Communication
    - Linux Internals : Interprocess Communication
- kunit
    - KUnit Testing Strategies
- Linux Media
    - v4l-utils.git - media (V4L2, DVB and IR) applications and libraries
    - drivers/media/test-drivers

    - https://www.linuxfoundation.org/webinars/testing-the-media-subsystem-compliance-tests-and-virtual-drivers?
    - https://elinux.org/Multimedia_Presentations
- Timers
    - https://docs.kernel.org/timers/timers-howto.html
    - Improving the kernel timers API [LWN.net]
    - High resolution timers and dynamic ticks design notes — The Linux Kernel documentation
- ALSA
    - Note: sound subsystem maintainers really don’t like codestyle cleanup patches, except the situations where the codestyle fixes go with other more sufficient changes.
    - Writing an ALSA Driver — The Linux Kernel documentation
    - Advanced Linux Sound Architecture - Driver Configuration guide
    - How PCM works in general (not only in Linux): https://www.xiph.org/video/
- SPI
    - Groking the Linux SPI Subsystem - Matt Porter, Konsulko
    - Serial Peripheral Interface (SPI) — The Linux Kernel documentation
- eBPF and Security
    - eBPF Observability Tools Are Not Security Tools
- File Systems
    - xfstests

## Linux Man Pages

- The Linux man-pages project
- Maintaining Linux man-pages
- https://git.kernel.org/pub/scm/docs/man-pages/man-pages.git/tree/CONTRIBUTING

## Linux kernel test rings

- https://linux.kernelci.org
- https://lkft.linaro.org/
- https://github.com/groeck/linux-build-test
- https://elinux.org/images/9/9f/Linux-Kernel-Testing-Where-are-we.pdf

## Common Weakness Enumerations

- The below view organizes weaknesses around concepts that are frequently used or encountered in software development. This includes all aspects of the software development lifecycle including both architecture and implementation. Accordingly, this view can align closely with the perspectives of architects, developers, educators, and assessment vendors. It provides a variety of categories that are intended to simplify navigation, browsing, and mapping.
    - CWE-699: Software Development (4.10)

## Browsing source

- cscope
- https://lxr.linux.no/linux+v6.0.9/Documentation/
- https://cregit.linuxsources.org/
- https://elixir.bootlin.com/linux/latest/source
- Tips for browsing source:
  - Look through code for 'todo' Additional source of todo: find linux/<your_subsytem> -name "*.c" | xargs grep -i todo
  - Architecture specific findindg. For example risc-v: ./Documentation/features/list-arch.sh riscv | grep TODO

## Mentee Shares

- https://prasad-udawant.gitbook.io/linux-kernel-development/toolchain-and-development-process/development-tools/code-review-and-submit-changes
- https://nixyogi.github.io/wiki/kernel/debugging/Reproducing-bugs-from-syzkaller.html
- https://nixyogi.github.io/wiki/kernel/debugging/Solving-syzkaller-bugs.html#process-for-solving-a-syzkaller-bug
- https://nixyogi.github.io/wiki/kernel/debugging/Solving-syzkaller-bugs.html
- A simple workflow to debug the Linux Kernel
- A better workflow for kernel debugging
- Booting a minimal upstream kernel in a Raspberry Pi 3
- The TTY Demystified
- Become a Linux kernel contributor - Part 3
- Learning eBPF by Liz Rice
- ZRAM: A Tool for Excessive Memory Use

## Conferences & Blogs

- http://retis.sssup.it/ospm-summit
- lwn.net
- Reader
- Dan Carpenter's Blog (mostly static analysis stuff)

## Delivering Presentations

- Best Practices for Mentees: How to Effectively Create and Deliver Presentations https://www.youtube.com/watch?v=2R8qt9DgXGg

## Current hot topics and patch opportunities

- Resource management API
  - DEFINE_FREE
  - no_free_ptr()
  - **[PATCH v3 00/57] Scope-based Resource Management - Peter Zijlstra**
  - **https://lore.kernel.org/all/2023061217-mutable-curry-c2ac@gregkh**
  - https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/include/linux/cleanup.h?h=v6.5-rc1