What kernel documentation could be

Jonathan Corbet
LWN.net
corbet@lwn.net
This is a discussion session!
Overall

We have a lot of documentation

Some of it is quite good

I think we can do better
The Linux Kernel documentation

This is the top level of the kernel's documentation tree. Kernel documentation, like the kernel itself, is very much a work in progress; that is especially true as we work to integrate our many scattered documents into a coherent whole. Please note that improvements to the documentation are welcome; join the linux-doc list at vger.kernel.org if you want to help out.

Licensing documentation

The following describes the license of the Linux kernel source code (GPLv2), how to properly mark the license of individual files in the source tree, as well as links to the full license text.

- Linux kernel licensing rules

User-oriented documentation

The following manuals are written for users of the kernel — those who are trying to get it to work optimally on a given system.

- The Linux kernel user’s and administrator’s guide
  - Linux kernel release 5.x <http://kernel.org/>
The Linux Kernel documentation

This is the top level of the kernel’s documentation tree. Kernel documentation, like the kernel itself, is very much a work in progress; that is especially true as we work to integrate our many scattered documents into a coherent whole. Please note that improvements to the documentation are welcome; join the documentation list at web.kernell.org if you want to help out.

Licensing documentation

The following describes the license of the Linux kernel source code (GPLv2), how to properly mark the license of individual files in the source tree, as well as links to the full license text.

- [Linux kernel licensing rules](#)

User-oriented documentation

The following manuals are written for users of the kernel — those who are trying to get it to work optimally on a given system.

- [The Linux kernel user’s and administrator’s guide](#)
  - [Linux kernel release 5.x](http://kernel.org/)
  - [The kernel’s command-line parameters](#)
The Linux Kernel documentation

This is the top level of the kernel’s documentation tree. Kernel documentation, like the kernel itself, is very much a work in progress; that is especially true as we work to integrate our many scattered documents into a coherent whole. Please note that improvements to the documentation are welcome; join the linux-doc list at vger.kernel.org if you want to help out.

Licensing documentation

The following describes the license of the Linux kernel source code (GPLv2), how to properly mark the license of individual files in the source tree, as well as links to the full license text.

- Linux kernel licensing rules

User-oriented documentation

The following manuals are written for users of the kernel — those who are trying to get it to work optimally on a given system.

- The Linux kernel user’s and administrator’s guide
  - Linux kernel release 5.x <http://kernel.org/>
  - The kernel’s command-line parameters
  - Linux allocated devices (4.x+ version)
  - Documentation for /proc/sys
  - Linux ABI description
  - Feature status on all architectures
  - Hardware vulnerabilities
  - Reporting issues
  - Reporting regressions
  - Security bugs
  - Bisecting a bug
  - Tainted kernel
  - Ramoops oops
Structure!
We *have* made some progress

admin-guide/
process/
core-api/
userspace-api/
driver-api/
maintainer/
...

The Linux Kernel documentation

This is the top level of the kernel's documentation tree. Kernel documentation, like the kernel itself, is very much a work in progress; that is especially true as we work to integrate our many scattered documents into a coherent whole. Please note that improvements to the documentation are welcome; join the linux-doc list at vger.kernel.org if you want to help out.

Working with the development community

The essential guides for interacting with the kernel's development community and getting your work upstream.

- A guide to the Kernel Development Process
- Submitting patches: the essential guide to getting your code into the kernel
- Code of conduct
- Kernel Maintainer Handbook
- All development-process docs

Internal API manuals

Manuals for use by developers working to interface with the rest of the kernel.

- Core API Documentation
- The Linux driver implementer's API guide
- Kernel subsystem API documentation
- Locking in the kernel

Development tools and processes

Various other manuals with useful information for all kernel developers.

- Linux kernel licensing rules
- How to write kernel documentation
- Development tools for the kernel
- Kernel Hacking Guides
- Linux Tracing Technologies
- fault-injection
- Kernel Livepatching

User-oriented documentation

The following manuals are written for users of the kernel — those who are trying to get it to work optimally on a given system.

- The Linux kernel user’s and administrator’s guide
- The kernel build system

Application-developer documentation
How do we impose more structure?

How do we fill in the gaps?
“Tribal knowledge”
Linux Memory Management Documentation

Memory Management Guide

This is a guide to understanding the memory management subsystem of Linux. If you are looking for advice on simply allocating memory, see the Memory Allocation Guide. For controlling and tuning guides, see the admin guide.

- Physical Memory
- Page Tables
- Process Addresses
- Boot Memory
- Page Allocation
- Virtually Contiguous Memory Allocation
- Slab Allocation
- High Memory Handling
- Page Reclaim
- Swap
- Page Cache
- Shared Memory Filesystem
- Out Of Memory Handling

Legacy Documentation

This is a collection of older documents about the Linux memory management (MM) subsystem internals with different level of details ranging from notes and mailing list responses for elaborating descriptions of data structures and algorithms. It should all be integrated nicely into the above structured documentation, or deleted
Virtually Contiguous Memory Allocation

© Copyright The kernel development community.
Built with Sphinx using a theme provided by Read the Docs.
process/submitting-patches.rst

It’s where we send new contributors

...838 lines of dense text ...
Translations

Chinese: active and growing
Traditional Chinese: stalled
Italian: maintained
Korean: abandoned
Japanese: abandoned (but coming back?)
Sphinx issues

Version support
currently 1.7..
Sphinx issues

Build times

<table>
<thead>
<tr>
<th>Version</th>
<th>Build Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.3</td>
<td>5:24</td>
</tr>
<tr>
<td>4.1.2</td>
<td>10:51</td>
</tr>
<tr>
<td>5.0.1</td>
<td>11:17</td>
</tr>
</tbody>
</table>
Sphinx issues

PDF generation
//! Synchronisation primitives.
//!
//! This module contains the kernel APIs related to synchronisation that have been ported
//! or wrapped for usage by Rust code in the kernel and is shared by all of them.
//!
//! # Example
//!
//! ```
//! # use kernel::mutex_init;
//! # use kernel::sync::Mutex;
//! # use alloc::boxed::Box;
//! # use core::pin::Pin;
//! // SAFETY: `init` is called below.
//! let mut data = Pin::from(Box::try_new(unsafe { Mutex::new(10) }).unwrap());
//! mutex_init!(data.as_mut(), "test::data");
//!
//! assert_eq!(*data.lock(), 10);
//! *data.lock() = 20;
//! assert_eq!(*data.lock(), 20);
//! ```
Rust!

Options:
  Require Rust code to use kernel-doc
  Find a way to incorporate rustdoc
$identifier = $1;
my $is_kernel_comment = 0;
my $decl_start = qr{$doc_com};
# test for pointer declaration type, foo * bar() - desc
my $fn_type = qr{\w+\s*\*\s*\s*};
my $parenthesis = qr{\(\w*\\)};
my $decl_end = qr{[-:].*};
if (/^$decl_start(\[\w\s\]+?)$parenthesis?\s*\$decl_end?$/) {
    $identifier = $1;
}
if ($identifier =~ m/^(struct|union|enum|typedef)\s*(\S*)/) {
    $decl_type = $1;
    $identifier = $2;
    $is_kernel_comment = 1;
}
# Look for foo() or static void foo() - description; or misspelt
# identifier
elsif (/^$decl_start$fn_type?\(\w+\)\s*\$parenthesis?\s*\$decl_end?$/) ||
    /^$decl_start$fn_type?\(\w+.*)$parenthesis?\s*\$decl_end$/) {
    $identifier = $1;
    $decl_type = 'function';
    $identifier =~ s/^define\s+//;
    $is_kernel_comment = 1;
}
$identifier =~ s/\s+$//;
What else?