

# 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



## Python 3.10.6 documentation

Welcome! This is the official documentation for Python 3.10.6.

### Parts of the documentation:

[What's new in Python 3.10?](#)

*or all "What's new" documents since 2.0*

[Tutorial](#)

*start here*

[Library Reference](#)

*keep this under your pillow*

[Language Reference](#)

*describes syntax and language elements*

[Python Setup and Usage](#)

*how to use Python on different platforms*

[Python HOWTOs](#)

*in-depth documents on specific topics*

[Installing Python Modules](#)

*installing from the Python Package Index & other sources*

[Distributing Python Modules](#)

*publishing modules for installation by others*

[Extending and Embedding](#)

*tutorial for C/C++ programmers*

[Python/C API](#)

*reference for C/C++ programmers*

[FAQs](#)

*frequently asked questions (with answers!)*

### Indices and tables:

[Global Module Index](#)

*quick access to all modules*

[General Index](#)

*all functions, classes, terms*

[Glossary](#)

*the most important terms explained*

[Search page](#)

*search this documentation*

[Complete Table of Contents](#)

*lists all sections and subsections*

### Meta information:

[Reporting bugs](#)

[Contributing to Docs](#)

[About the documentation](#)

[History and License of Python](#)

[Copyright](#)

© 2001-2022 Python Software Foundation.  
This page is licensed under the Python Software Foundation License Version 2.  
Examples, recipes, and other code in the documentation are additionally licensed under the Zero Clause BSD License.  
See [History and License](#) for more information.

The Python Software Foundation is a non-profit corporation. [Please donate.](#)

Last updated on Aug 26, 2022. [Found a bug?](#)

Created using [Sphinx](#) 3.4.3.



## Python 3.10.6 documentation

Welcome! This is the official documentation for Python 3.10.6.

### Parts of the documentation:

[What's new in Python 3.10?](#)  
*or all "What's new" documents since 2.0*

[Tutorial](#)  
*start here*

[Library Reference](#)  
*keep this under your pillow*

[Language Reference](#)  
*describes syntax and language elements*

[Python Setup and Usage](#)  
*how to use Python on different platforms*

[Python HOWTOs](#)  
*in-depth documents on specific topics*

[Installing Python Modules](#)  
*installing from the Python Package Index & other sources*

[Distributing Python Modules](#)  
*publishing modules for installation by others*

[Extending and Embedding](#)  
*tutorial for C/C++ programmers*

[Python/C API](#)  
*reference for C/C++ programmers*

[FAQs](#)  
*frequently asked questions (with answers!)*

### Indices and tables:

[Global Module Index](#)  
*quick access to all modules*

[General Index](#)  
*all functions, classes, terms*

[Glossary](#)  
*the most important terms explained*

[Search page](#)  
*search this documentation*

[Complete Table of Contents](#)  
*lists all sections and subsections*

### Meta information:

[Reporting bugs](#)

[Contributing to Docs](#)

[About the documentation](#)

[History and License of Python](#)

[Copyright](#)

This page is licensed under the Python Software Foundation License Version 2. Examples, recipes, and other code in the documentation are additionally licensed under the Zero Clause BSD License. See [History and License](#) for more information.

The Python Software Foundation is a non-profit corporation. [Please donate.](#)

Last updated on Aug 26, 2022. [Found a bug?](#)  
Created using [Sphinx](#) 3.4.3.

## 🏠 The Linux Kernel

6.0.0-rc2

Search docs

The Linux kernel user's and administrator's guide

Kernel Build System

The Linux kernel firmware guide

Open Firmware and Devicetree

The Linux kernel user-space API guide

Working with the kernel development community

Development tools for the kernel

How to write kernel documentation

Kernel Hacking Guides

Linux Tracing Technologies

Kernel Maintainer Handbook

fault-injection

Kernel Livepatching

The Linux driver implementer's API guide

Core API Documentation

locking

Accounting

Block

cdrom

Linux CPUFreq - CPU frequency

## 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](mailto: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/>](http://kernel.org/)
  - [The kernel's command-line parameters](#)

## Python 3.10.6 documentation

Welcome! This is the official documentation for Python 3.10.6.

### Parts of the documentation:

- [What's new in Python 3.10?](#)  
*or all "What's new" documents since 2.0*
- [Tutorial](#)  
*start here*
- [Library Reference](#)  
*keep this under your pillow*
- [Language Reference](#)  
*describes syntax and language elements*
- [Python Setup and Usage](#)  
*how to use Python on different platforms*
- [Python HOWTOs](#)  
*in-depth documents on specific topics*

- [Installing Python Modules](#)  
*installing from the Python Package Index & other sources*
- [Distributing Python Modules](#)  
*publishing modules for installation by others*
- [Extending and Embedding](#)  
*tutorial for C/C++ programmers*
- [Python/C API](#)  
*reference for C/C++ programmers*
- [FAQs](#)  
*frequently asked questions (with answers!)*

### Indices and tables:

- [Global Module Index](#)  
*quick access to all modules*
- [General Index](#)  
*all functions, classes, terms*
- [Glossary](#)  
*the most important terms explained*

- [Search page](#)  
*search this documentation*
- [Complete Table of Contents](#)  
*lists all sections and subsections*

### Meta information:

- [Reporting bugs](#)
- [Contributing to Docs](#)
- [About the documentation](#)

- [History and License of Python](#)
- [Copyright](#)

This page is licensed under the Python Software Foundation License Version 2. Examples, recipes, and other code in the documentation are additionally licensed under the Zero Clause BSD License. See [History and License](#) for more information.

The Python Software Foundation is a non-profit corporation. [Please donate.](#)

Last updated on Aug 26, 2022. [Found a bug?](#)  
Created using [Sphinx](#) 3.4.3.

- The Linux kernel user's and administrator's guide
- Kernel Build System
- The Linux kernel firmware guide
- Open Firmware and Devicetree
- The Linux kernel user-space API guide
- Working with the kernel development community
- Development tools for the kernel
- How to write kernel documentation
- Kernel Hacking Guides
- Linux Tracing Technologies
- Kernel Maintainer Handbook
- fault-injection
- Kernel Livepatching
- The Linux driver implementer's API guide
- Core API Documentation
- locking
- Accounting
- Block
- cdrom
- Linux CPUFreq - CPU frequency

# 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 our [linux-doc](#) list at [vmcs@kernel.org](mailto:vmcs@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](#)

» The Linux Kernel documentation

## 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](mailto: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/>](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](#)
  - [Bug hunting](#)
  - [Bisecting a bug](#)
  - [Tainted kernel](#)
  - [Ramoops oop](#)

**Print**

26 sheets of paper

Destination

Save to PDF

Orientation

Portrait Landscape

Pages

All

Color mode

Color

Fewer settings

Paper size

US Letter

Scale

 Fit to page width Scale 100

Pages per sheet

1

Cancel

Save



# Structure!







*We have* made some progress

admin-guide/

process/

core-api/

userspace-api/

driver-api/

maintainer/

...



## Quick search

## Table of Contents

- The Linux Kernel documentation
  - Working with the development community
  - Internal API manuals
  - Development tools and processes
  - User-oriented documentation
  - Application-developer documentation
  - Firmware-related documentation
  - Architecture-agnostic documentation
  - Architecture-specific documentation
  - Other documentation
    - Translations
- Indices and tables

## This Page

[Show Source](#)

# 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](mailto: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



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

BPF Documentation

USB support

Linux PCI Bus Subsystem

Linux SCSI Subsystem

Assorted Miscellaneous Devices Documentation

Linux Scheduler

MHI

Linux PECCI Subsystem

Assembler Annotations

CPU Architectures

Kernel tools

Unsorted Documentation

Atomic Types

Atomic bitops

Memory Barriers

Translations

[🏠](#) » [Linux Memory Management Documentation](#) »

Virtually Contiguous Memory Allocation

[View page source](#)

---

## Virtually Contiguous Memory Allocation

[← Previous](#)

[Next →](#)

---

© 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

2.4.3	5:24
4.1.2	10:51
5.0.1	11:17



# Sphinx issues

PDF generation



# Rust!



```
//! 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!

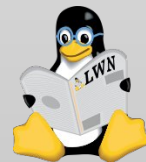
## Options:

- Require Rust code to use kernel-doc
- Find a way to incorporate rustdoc



# kernel-doc

```
$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*};
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)\b\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?

