Kernel documentation

Jonathan Corbet LWN.net corbet@lwn.net



Some 2023 developments

Arch docs move to Documentation/arch Integration of Rust docs Spanish translations Switch to the "Alabaster" theme Rendering docs into the texinfo format Rewritten top-level index.rst



Lots of docs written!



What next?



Create Documentation/devices Put device-related docs there (~31 subdirectories)



Structure!



Plain text v. HTML



Stats from docs.kernel.org

150,000 requests/day Top docs: process/coding-style process/submitting-patches process/maintainer-netdev admin-guide/kernel-parameters admin-guide/reporting-issues core-api/kernel-api driver-api/basics



Tooling

Sphinx:

Docs-build speed

Minimum version to support?

Doing 1.7 -> 2.4, go further?

Better partial-build support

Better sidebar

Working with the Sphinx community



All development-process docs	Working with the kernel development	
Linux kernel licensing	community	
rules		
HOWTO do Linux kernel development Contributor Covenant Code of Conduct	So you want to be a Linux kernel developer? Welcome! While there is a lot to be learned about the kernel in a technical sense, it is also important to learn about how our community works. Reading these documents will make it much easier for you to get your changes merged with a minimum of trouble.	
Linux Kernel Contributor	Below are the essential guides that every developer should read.	
Covenant Code of	below are the essential guides that every developer should read.	
Conduct Interpretation	Linux kernel licensing rules	
A guide to the Kernel	HOWTO do Linux kernel development	
Development Process	Contributor Covenant Code of Conduct	
Submitting patches: the	• Linux Kernel Contributor Covenant Code of Conduct Interpretation	
essential guide to	A guide to the Kernel Development Process	
getting your code into	• Submitting patches: the essential guide to getting your code into the kernel	
the kernel	Handling regressions	
Handling regressions	Programming Language	
Programming Language	Linux kernel coding style	
Linux kernel coding style	Subsystem and maintainer tree specific development process notes	
Subsystem and maintainer	Kernel Maintainer PGP guide	
tree specific	• Email clients info for Linux	
development process	• Linux Kernel Enforcement Statement	
notes	Kernel Driver Statement	
Kernel Maintainer PGP guide	For security issues, see:	
Email clients info for	Security bugs	
Linux	Embargoed hardware issues	
Linux Kernel Enforcement Statement	Other guides to the community that are of interest to most developers are:	
Kernel Driver Statement	Minimal requirements to compile the Kernel	
Security bugs	• The Linux Kernel Driver Interface	
Embargoed hardware	Linux kernel management style	
issues	• Everything you ever wanted to know about Linux -stable releases	
Minimal requirements to	• Linux Kernel patch submission checklist	LIWA
compile the Kernel	Index of Further Kernel Documentation	
The Linux Kernel Driver	Deprecated Interfaces, Language Features, Attributes, and Conventions	
Interface	• List of maintainers	

All development-process Working with the kernel development docs community Linux kernel licensing rules So you want to be a Linux kernel developer? Welcome! While there is a lot to be HOWTO do Linux kernel learned about the kernel in a technical sense, it is also important to learn about how development our community works. Reading these documents will make it much easier for you to Contributor Covenant get your changes merged with a minimum of trouble. Code of Conduct Linux Kernel Contributor Below are the essential guides that every developer should read. Covenant Code of Linux kernel licensing rules Conduct Interpretation • HOWTO do Linux kernel development A guide to the Kernel · Contributor Covenant Code of Conduct **Development Process** • Linux Kernel Contributor Covenant Code of Conduct Interpretation Submitting patches: the • A guide to the Kernel Development Process essential guide to • Submitting patches: the essential guide to getting your code into the kernel getting your code into · Handling regressions the kernel • Programming Language Handling regressions · Linux kernel coding style Programming Language • Subsystem and maintainer tree specific development process notes Linux kernel coding style · Kernel Maintainer PGP guide Subsystem and maintainer · Email clients info for Linux tree specific · Linux Kernel Enforcement Statement development process Kernel Driver Statement notes Kernel Maintainer PGP For security issues, see: quide Security bugs Email clients info for · Embargoed hardware issues Linux Linux Kernel Enforcement Other guides to the community that are of interest to most developers are: Statement Kernel Driver Statement Minimal requirements to compile the Kernel Security bugs • The Linux Kernel Driver Interface Embargoed hardware · Linux kernel management style issues • Everything you ever wanted to know about Linux -stable releases Minimal requirements to · Linux Kernel patch submission checklist compile the Kernel • Index of Further Kernel Documentation The Linux Kernel Driver Deprecated Interfaces, Language Features, Attributes, and Conventions Interface List of maintainers

All development-process Working with the kernel development docs community Linux kernel licensing rules So you want to be a Linux kernel developer? Welcome! While there is a lot to be HOWTO do Linux kernel learned about the kernel in a technical sense, it is also important to learn about how development our community works. Reading these documents will make it much easier for you to Contributor Covenant get your changes merged with a minimum of trouble. Code of Conduct Linux Kernel Contributor Below are the essential guides that every developer should read. Covenant Code of · Linux kernel licensing rules Conduct Interpretation • HOWTO do Linux kernel development A guide to the Kernel · Contributor Covenant Code of Conduct **Development Process** • Linux Kernel Contributor Covenant Code of Conduct Interpretation Submitting patches: the • A guide to the Kernel Development Process essential guide to • Submitting patches: the essential guide to getting your code into the kernel getting your code into · Handling regressions the kernel • Programming Language Handling regressions · Linux kernel coding style Programming Language • Subsystem and maintainer tree specific development process notes Linux kernel coding style · Kernel Maintainer PGP guide Subsystem and maintainer · Email clients info for Linux tree specific · Linux Kernel Enforcement Statement development process Kernel Driver Statement notes Kernel Maintainer PGP For security issues, see: quide Security bugs Email clients info for · Embargoed hardware issues Linux Linux Kernel Enforcement Other guides to the community that are of interest to most developers are: Statement Kernel Driver Statement Minimal requirements to compile the Kernel Security bugs • The Linux Kernel Driver Interface Embargoed hardware · Linux kernel management style issues • Everything you ever wanted to know about Linux -stable releases Minimal requirements to · Linux Kernel patch submission checklist compile the Kernel • Index of Further Kernel Documentation The Linux Kernel Driver Deprecated Interfaces, Language Features, Attributes, and Conventions Interface List of maintainers

All development-process Working with the kernel development docs community Linux kernel licensing rules So you want to be a Linux kernel developer? Welcome! While there is a lot to be HOWTO do Linux kernel learned about the kernel in a technical sense, it is also important to learn about how development our community works. Reading these documents will make it much easier for you to Contributor Covenant get your changes merged with a minimum of trouble. Code of Conduct Linux Kernel Contributor Below are the essential guides that every developer should read. Covenant Code of · Linux kernel licensing rules Conduct Interpretation • HOWTO do Linux kernel development A guide to the Kernel · Contributor Covenant Code of Conduct **Development Process** • Linux Kernel Contributor Covenant Code of Conduct Interpretation Submitting patches: the • A guide to the Kernel Development Process essential guide to • Submitting patches: the essential guide to getting your code into the kernel getting your code into · Handling regressions the kernel • Programming Language Handling regressions · Linux kernel coding style Programming Language • Subsystem and maintainer tree specific development process notes Linux kernel coding style • Kernel Maintainer PGP guide Subsystem and maintainer · Email clients info for Linux tree specific • Linux Kernel Enforcement Statement development process Kernel Driver Statement notes Kernel Maintainer PGP For security issues, see: quide Security bugs Email clients info for · Embargoed hardware issues Linux Linux Kernel Enforcement Other guides to the community that are of interest to most developers are: Statement Kernel Driver Statement Minimal requirements to compile the Kernel Security bugs • The Linux Kernel Driver Interface Embargoed hardware · Linux kernel management style issues • Everything you ever wanted to know about Linux -stable releases Minimal requirements to · Linux Kernel patch submission checklist compile the Kernel • Index of Further Kernel Documentation The Linux Kernel Driver Deprecated Interfaces, Language Features, Attributes, and Conventions Interface List of maintainers

All development-process docs	Working with the kernel development	
HOWTO do Linux kernel	community	
development	Community	
A guide to the Kernel	So you want to be a Linux kernel developer? Welcome! While there is a lot to be learned about the kernel in a technical sense, it is also important to learn about how	
Development Process Submitting patches: the	our community works. Reading these documents will make it much easier for you to	
essential guide to	get your changes merged with a minimum of trouble.	
getting your code into the kernel	Section 1: an introduction to how kernel development works	
Linux Kernel patch	HOWTO do Linux kernel development	
submission checklist	• A guide to the Kernel Development Process	
Minimal requirements to	 Submitting patches: the essential guide to getting your code into the kernel 	
compile the Kernel	Linux Kernel patch submission checklist	
Programming Language	Section 2: Tools and technical guides for kernel developers	
Linux kernel coding style	Section 2. 10013 and technical guides for kernel developers	
Kernel Maintainer PGP	Minimal requirements to compile the Kernel	
guide	Programming Language	
Email clients info for	Linux kernel coding style	
Linux	Kernel Maintainer PGP guide	
Applying Patches To The	Email clients info for Linux	
Linux Kernel	Applying Patches To The Linux Kernel	
Backporting and conflict	Backporting and conflict resolution	
resolution	Adding a New System Call	
Adding a New System Call	Why the "volatile" type class should not be used	
Why the "volatile" type	• (How to avoid) Botching up ioctls	
class should not be used	Caption 2. Delice guides and developen statements, these are the miles that are true to	
(How to avoid) Botching	Section 3 : Policy guides and developer statements: these are the rules that we try to live by in the kernel community (and beyond).	
up ioctls	nve by in the kerner community (and beyond).	
Linux kernel licensing	Linux kernel licensing rules	
rules	Contributor Covenant Code of Conduct	
Contributor Covenant	• Linux Kernel Contributor Covenant Code of Conduct Interpretation	
Code of Conduct	Linux Kernel Contribution Maturity Model	
Linux Kernel Contributor	Linux Kernel Enforcement Statement	
Covenant Code of	Kernel Driver Statement	LWN
Conduct Interpretation	The Linux Kernel Driver Interface	
Linux Kernel Contribution	 Everything you ever wanted to know about Linux -stable releases 	
Maturity Model	Linux kernel management style	

How do we support this work?

