Kernel documentation: what we have and where we're going

Jonathan Corbet
corbet@lwn.net

Kernel Recipes 2016

> Maintainers tend to get to be maintainers because they were good at something else, and not good enough at hiding from the "maintainer" role. There is a paradox here as a maintainer must be good at saying "No", but if they were they might never have agreed to become a maintainer.
> — Neil Brown

The current state of kernel documentation

The Linux kernel

The core of any Linux system

Some numbers:

- 53,654 files
- 3,689 directories
- 63 -day release cycle (+/-)
- 1,600 developers contributing to each release
- 12,000 changesets (at least) in each release

A huge and fast-moving project!

Kernel documentation

Centered in `Documentation/`

- 2,264 files
  - 228 directories
- 23MB of material

(all excluding `Documentation/devicetree`)

Also some utilities in `scripts/`

...and arguably stuff in `samples/`

The two faces of kernel documentation

- Ordinary .txt files (2000+ of them)
  - Some are current, comprehensive, and useful
  - Others … less so...

The two faces of kernel documentation

- Ordinary .txt files (2000+ of them)
  - Some are current, comprehensive, and useful
  - Others … less so...
- Formatted documentation under DocBook/
  - 34 DocBook “template files”
  - Can pull information from the source code
  - Rendered into PDF, HTML, man pages, ...

But wait … there's more …!

Kerneldoc comments

Found throughout the kernel source

```
/**
 * list_add - add a new entry
 * @new: new entry to be added
 * @head: list head to add it after
 *
 * Insert a new entry after the specified head.
 * This is good for implementing stacks.
 */
```

Kerneldoc comments

Found throughout the kernel source

```
/**
 * list_add - add a new entry
 * @new: new entry to be added
 * @head: list head to add it after
 *
 * Insert a new entry after the specified head.
 * This is good for implementing stacks.
 */
```

Kerneldoc comments

Found throughout the kernel source

```
/**
 * list_add - add a new entry
 * @new: new entry to be added
 * @head: list head to add it after
 *
 * Insert a new entry after the specified head.
 * This is good for implementing stacks.
 */
```

Kerneldoc comments

Found throughout the kernel source

```
/**
 * list_add - add a new entry
 * @new: new entry to be added
 * @head: list head to add it after
 *
 * Insert a new entry after the specified head.
 * This is good for implementing stacks.
 */
```

Kerneldoc comments

Found throughout the kernel source

```
/**
 * list_add - add a new entry
 * @new: new entry to be added
 * @head: list head to add it after
 *
 * Insert a new entry after the specified head.
 * This is good for implementing stacks.
 */
```

Kerneldoc comments

Can describe structures as well as functions

Can also contain unattached documentation

The kernel contains about 55,000 of them!

Intelligent design?

There is no overall vision for kernel docs

- Things get added whenever...
- ...and wherever...
- ...and forgotten for years.

“Documentation/* is a gigantic mess, currently organized based on where random passers-by put things down last.”
— Rob Landley

“Documentation/* is a gigantic mess, currently organized based on where random passers-by put things down last.”
— Rob Landley, July, 2007

Terminal

File Edit View Search Terminal Help

```
70 xps:...kernel/Documentation | d
00-INDEX                    devices.txt              io_ordering.txt          module-signing.txt       serial/
ABI/                        devicetree/              iostats.txt              mono.txt                 serial-console.txt
accounting/                 digsig.txt               IPMI.txt                 mtd/                     sgi-ioc4.txt
acpi/                       DMA-API-HOWTO.txt        IRQ-affinity.txt         namespaces/              sh/
adding-syscalls.txt         DMA-API.txt              IRQ-domain.txt           netlabel/                SM501.txt
aoe/                        DMA-attributes.txt       irqflags-tracing.txt     networking/              smsc_ece1099.txt
applying-patches.txt        dma-buf-sharing.txt      IRQ.txt                  nfc/                     sound/
arm/                        dmaengine/               isapnp.txt               nios2/                   sparse.txt
arm64/                      DMA-ISA-LPC.txt          isdn/                    nommu-mmap.txt           spi/
assoc_array.txt             DocBook/                 ja_JP/                   ntb.txt                  stable_api_nonsense.txt
atomic_ops.txt              dontdiff                 java.txt                 numastat.txt             stable_kernel_rules.txt
auxdisplay/                 driver-model/            kasan.txt                nvdimm/                  static-keys.txt
backlight/                  dvb/                     kbuild/                  nvmem/                   SubmitChecklist
bad_memory.txt              dynamic-debug-howto.txt  kdump/                   oops-tracing.txt         SubmittingDrivers
basic_profiling.txt         early-userspace/         kernel-doc-nano-HOWTO.txt padata.txt              SubmittingPatches
bcache.txt                  edac.txt                 kernel-docs.txt          parisc/                  svga.txt
binfmt_misc.txt             EDID/                    kernel-parameters.txt    parport-lowlevel.txt     sysctl/
blackfin/                   efi-stub.txt             kernel-per-CPU-kthreads.txt parport.txt           sysfs-rules.txt
block/                      eisa.txt                 kmemcheck.txt            PCI/                     sysrq.txt
blockdev/                   email-clients.txt        kmemleak.txt             pcmcia/                  target/
braille-console.txt         extcon/                  kobject.txt              percpu-rw-semaphore.txt  thermal/
bt8xxgpio.txt               fault-injection/         ko_KR/                   phy/                     this_cpu_ops.txt
btmrvl.txt                  fb/                      kprobes.txt              phy.txt                  timers/
BUG-HUNTING                 features/                kref.txt                 pi-futex.txt             tpm/
bus-devices/                filesystems/             kselftest.txt            pinctrl.txt              trace/
bus-virt-phys-mapping.txt   firmware_class/         laptops/                 platform/                unaligned-memory-access.txt
cachetlb.txt                flexible-arrays.txt      ldm.txt                  pnp.txt                  unicode.txt
cdrom/                      fmc/                     leds/                    power/                   unshare.txt
cgroups/                    fpga/                    local_ops.txt            powerpc/                 usb/
Changes                     frv/                     locking/                 pps/                     vDSO/
circular-buffers.txt        futex-requeue-pi.txt     lockup-watchdogs.txt     prctl/                   vfio.txt
clk.txt                     gcov.txt                 logo.gif                 preempt-locking.txt      vgaarbiter.txt
cma/                        gdb-kernel-debugging.txt logo.txt                 printk-formats.txt       VGA-softcursor.txt
coccinelle.txt              gpio/                    lzo.txt                  pti/                     video4linux/
CodeOfConflict              hid/                     m68k/                    ptp/                     video-output.txt
CodingStyle                 highuid.txt              magic-number.txt         pwm.txt                  virtual/
connector/                  HOWTO                    mailbox.txt              ramoops.txt              vm/
console/                    hsi.txt                  Makefile                 rapidio/                 vme_api.txt
cpu-freq/                   hwmon/                   ManagementStyle          rbtree.txt               volatile-considered-harmful.txt
cpu-hotplug.txt             hw_random.txt            md-cluster.txt           RCU/                     w1/
cpuidle/                    hwspinlock.txt           md.txt                   remoteproc.txt           watchdog/
cpu-load.txt                i2c/                     media-framework.txt      rfkill.txt               wimax/
cputopology.txt             ia64/                    memory-barriers.txt      robust-futex-ABI.txt     workqueue.txt
crc32.txt                   ide/                     memory-devices/          robust-futexes.txt       x86/
cris/                       infiniband/              memory-hotplug.txt       rpmsg.txt                xillybus.txt
crypto/                     initrd.txt               men-chameleon-bus.txt    rtc.txt                  xtensa/
dcdbas.txt                  init.txt                 metag/                   s390/                    xz.txt
debugging-modules.txt       input/                   mic/                     SAK.txt                  zh_CN/
debugging-via-ohci1394.txt  Intel-IOMMU.txt          mips/                    scheduler/               zorro.txt
dell_rbu.txt                intel_txt.txt            misc-devices/            scsi/
development-process/        ioctl/                   mmc/                     security/
device-mapper/              io-mapping.txt           mn10300/                 SecurityBugs
71 xps:...kernel/Documentation |
```

Intelligent design?

There is no overall vision for kernel docs
- Things get added whenever...
- ...and wherever...
- ...and forgotten for years.

No cross-document linkage
- Few references to other documents at all

Formatted documentation

The real heart of kernel documentation?

Advantages:

- Much of the documentation is in the source
- Enables creation of integrated documents
  - ...to a point
- Numerous output formats
- Active interest in improving things

There's a “but” coming...

It's a documentation system written by kernel developers!

How it works

User types “make htmldocs”

scripts/docproc parses the template files

```
<sect1><title>String Conversions</title>
!Elib/vsprintf.c
!Finclude/linux/kernel.h kstrtol
!Finclude/linux/kernel.h kstrtoul
!Elib/kstrtox.c
```

How it works

For each source-code reference:

- docproc parses the source file for exports

How it works

For each source-code reference:

- docproc parses the source file for exports
- kernel-doc parses the source for definitions

How it works

For each source-code reference:

- docproc parses the source file for exports
- kernel-doc parses the source for definitions
- kernel-doc parses it *again* for documentation
  - emits DocBook-formatted snippets

How it works

For each source-code reference:

- docproc parses the source file for exports
- kernel-doc parses the source for definitions
- kernel-doc parses it *again* for documentation
  - emits DocBook-formatted snippets
- docproc stuffs them into the template

How it works

For each source-code reference:

- docproc parses the source file for exports
- kernel-doc parses the source for definitions
- kernel-doc parses it *again* for documentation
  - emits DocBook-formatted snippets
- docproc stuffs them into the template
- kernel-doc-xml-ref finds cross-references
  - only within a single template!
  - template munged appropriately

How it works

For each source-code reference:

- docproc parses the source file for exports
- kernel-doc parses the source for definitions
- kernel-doc parses it *again* for documentation
  - emits DocBook-formatted snippets
- docproc stuffs them into the template
- kernel-doc-xml-ref finds cross-references
  - only within a single template!
  - template munged appropriately
- xmlto is fed the result

Resulting problems

It's slow

It's brittle

Hard to set up and make work

Ugly output

No formatting within DocBook comments

No integration with the rest of the `Documentation/` directory

How can we make things better?

Toward better kernel docs

Clean up the mess

Better formatted output

- With a more rational toolchain

Preserve — or enhance — plain-text access

Recent work

Add markdown processing to kerneldoc comments

(Daniel Vetter, Daniel Cesar Lemes de Paula)

Later switched to asciidoc

AsciiDoc in kernel comments

Advantages

- Move more documentation to the source
- Avoid unpleasant DocBook formatting
- Better documentation!

AsciiDoc in kernel comments

Advantages

- Move more documentation to the source
- Avoid unpleasant DocBook formatting
- Better documentation!

Disadvantages

- Adding a new tool to the house of cards
- Disagreements between tools
- Performance issues
- Still no linkage between documents
- Ruby dependency

What I wanted to see

Dispense with DocBook entirely!

Use a simple markup processor for everything
(Markdown, AsciiDoc, Sphinx, …)
Even for “unformatted” docs

Create a single integrated documentation tree

Working solutions should not be blocked in favor of something nicer that might exist in the future.

...even so...

...one can look around a *little* more...

Sphinx

A documentation system for Python

- www.sphinx-doc.org

Based on reStructuredText

- Yet another simple markup language

Advantages of Sphinx

- Designed for documenting code
- Designed for large documents
  - ...in multiple files
- Widely used, well supported
- Output to HTML, ePub, PDF
  - ...without DocBook or even LaTeX

What happened from there

I posted a proof of concept
  discussion ensued

Jani Nikula took the POC and ran with it

Consensus formed around this approach

How it works

The kerneldoc comments work as always

- No need to change 1000’s of comments

But...

- Those comments can now contain RST directives!

We also get:

Cross-document cross-references

Function / data structure indexes

Much nicer output

Simpler, faster document build

Including kernel-doc comments

The Linux kernel source files may contain structured documentation comments, or kernel-doc comments to describe the functions and types and design of the code. The documentation comments may be included to any of the reStructuredText documents using a dedicated kernel-doc Sphinx directive extension.

The kernel-doc directive is of the format:

```
.. kernel-doc:: source
   :option:
```

The *source* is the path to a source file, relative to the kernel source tree. The following directive options are supported:

export: ***[source-pattern ...]***

Include documentation for all functions in *source* that have been exported using `EXPORT_SYMBOL` or `EXPORT_SYMBOL_GPL` either in *source* or in any of the files specified by *source-pattern*.

The *source-pattern* is useful when the kernel-doc comments have been placed in header files, while `EXPORT_SYMBOL` and `EXPORT_SYMBOL_GPL` are next to the function definitions.

Examples:

```
.. kernel-doc:: lib/bitmap.c
```

How it works (continued)

- Place an RST document in Documentation/
  - Doesn’t have to be in a specific directory
- Add a reference in `index.rst`
- That’s it!

kernel: All files - gitk

File Edit View Help

docs: Sphinxify gdb-kernel-debugging.txt and move to dev-tools	Jonathan Corbet <corbet@lwn.net>	2016-08-08 15:55:49
docs: sphinxify kmemcheck.txt and move to dev-tools	Jonathan Corbet <corbet@lwn.net>	2016-08-07 16:12:28
docs: sphinxify kmemleak.txt and move it to dev-tools	Jonathan Corbet <corbet@lwn.net>	2016-08-07 15:46:10
docs: sphinxify ubsan.txt and move it to dev-tools	Jonathan Corbet <corbet@lwn.net>	2016-08-07 15:35:42
docs: sphinxify kasan.txt and move to dev-tools	Jonathan Corbet <corbet@lwn.net>	2016-08-07 15:31:03
docs: sphinixfy gcov.txt and move to dev-tools	Jonathan Corbet <corbet@lwn.net>	2016-08-07 15:26:20
docs: sphinxify kcov.txt and move to dev-tools	Jonathan Corbet <corbet@lwn.net>	2016-08-07 15:13:00
docs: sphinxify sparse.txt and move to dev-tools	Jonathan Corbet <corbet@lwn.net>	2016-08-07 15:09:14
docs: sphinxify coccinelle.txt and add it to dev-tools	Jonathan Corbet <corbet@lwn.net>	2016-08-08 16:03:14
docs: create a new dev-tools directory	Jonathan Corbet <corbet@lwn.net>	2016-08-08 16:00:25

SHA1 ID: 4b9033a33494ec9154d63e706e9e47f7eb3fd59e Row 90/ 616142

Find commit containing: Exact All fields

Search

Diff Old version New version Lines of context: 3 Ignore space change Line diff

```
 positives. Thus, reports must be carefully checked, and patches
 reviewed.

-To enable verbose messages set the V= variable, for example:
+To enable verbose messages set the V= variable, for example::

    make coccicheck MODE=report V=1

- Coccinelle parallelization
-~~~~~~~~~~~~~~~~~~~~~~~~~~
+Coccinelle parallelization
+--------------------------

 By default, coccicheck tries to run as parallel as possible. To change
-the parallelism, set the J= variable. For example, to run across 4 CPUs:
+the parallelism, set the J= variable. For example, to run across 4 CPUs::

    make coccicheck MODE=report J=4

@@ -115,44 +125,47 @@ As of Coccinelle 1.0.2 Coccinelle uses Ocaml parmap for parallelizat
 if support for this is detected you will benefit from parmap parallelization.

 When parmap is enabled coccicheck will enable dynamic load balancing by using
-'--chunksize 1' argument, this ensures we keep feeding threads with work
+``--chunksize 1`` argument, this ensures we keep feeding threads with work
 one by one, so that we avoid the situation where most work gets done by only
 a few threads. With dynamic load balancing, if a thread finishes early we keep
 feeding it more work.
```

Patch Tree

Comments
Documentation/coccinelle.txt
Documentation/dev-tools/coccinelle.rst
Documentation/dev-tools/tools.rst
MAINTAINERS

Including kerneldoc comments

Jani wrote a new extension module

Use the following:

```
.. kernel-doc:: file
   :export:
   :internal:
   :doc: doc-section title
   :function: functions ...
```

Including kerneldoc comments

Jani wrote a new extension module

Use the following:

```
.. kernel-doc:: file
   :export:
   :internal:
   :doc: doc-section title
   :function: functions ...
```

Only one of these!

Current status

Merged for 4.8

Sphinx documents for now

- Kernel documentation HOWTO
- GPU documentation
- Media subsystem

The merge window has been fairly normal, although the patch itself looks somewhat unusual: over 20% of the patch is documentation updates, due to conversion of the drm and media documentation from docbook to the Sphinx doc format.
— Linus Torvalds (4.8-rc1 release)

My general impression is that it is now a way easier to maintain the media documentation and make it more consistent than with DocBook.
— Mauro Carvalho Chehab

This new documentation format combines the best of two worlds, pretty online browser documentation with almost plain text files, and changes being tracked via git commits.... You got to love it! :-)
— Jesper Dangaard Brouer

Coming in 4.9

Documentation/driver-api/

- Conversion of the device-drivers book
  (+ various text files)

Documentation/driver-api

Start with DocBook/device-drivers.tmpl

- Convert to several .rst files under driver-api/

Documentation/driver-api

Start with DocBook/device-drivers.tmpl

- Convert to several .rst files under driver-api/

Grab Documentation/hsi.txt

- Merge with driver-api/hsi.rst
- HSI is now documented in one place!

Documentation/driver-api

Start with DocBook/device-drivers.tmpl
- Convert to several .rst files under driver-api/

Grab Documentation/hsi.txt
- Merge with driver-api/hsi.rst
- HSI is now documented in one place!

(This is just the beginning)

Coming in 4.9

Documentation/driver-api/

- Conversion of the device-drivers book (+ various text files)

Documentation/dev-tools/

- Coalesced tool documents in RST

PDF output

- Painful, with LaTex

4.10?

Documentation/process/

- HOWTO
- CodingStyle
- SubmittingPatches
- stable_kernel_rules.txt
- ManagementStyle
- development-process
- ...

Future work

- Convert other DocBook documents
  - Eventually eliminate DocBook altogether
- Rethink the kernel-doc utility
  - 20 years of Perl cruft!
- Incorporate more plain-text documents
- Bring some order to Documentation/

Future work

Convert other DocBook documents
- Eventually eliminate DocBook altogether

Rethink the kernel-doc utility
- 20 years of Perl cruft!

Incorporate more plain-text documents

Bring some order to Documentation/

Create more, better documentation!

Questions / thoughts?

> I've mentioned that the new toolchain is awesome, right?
> — Daniel Vetter