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

Kernel Recipes 2019

Jonathan Corbet
corbet@lwn.net

# Why does documentation matter?

A crucial aid to our users

An aid for our developers

It makes us think about what we’re doing

Documentation is a key to building a healthy community

# The Linux kernel

The core of any Linux system

Some numbers:

- 68,000 files
- 5,000 directories
- 63-70 day release cycle (+/-)
- 1,700 developers contributing to each release
  (>4000 over the course of a year)
- 13,000 changesets (at least) in each release

# A huge and fast-moving project!

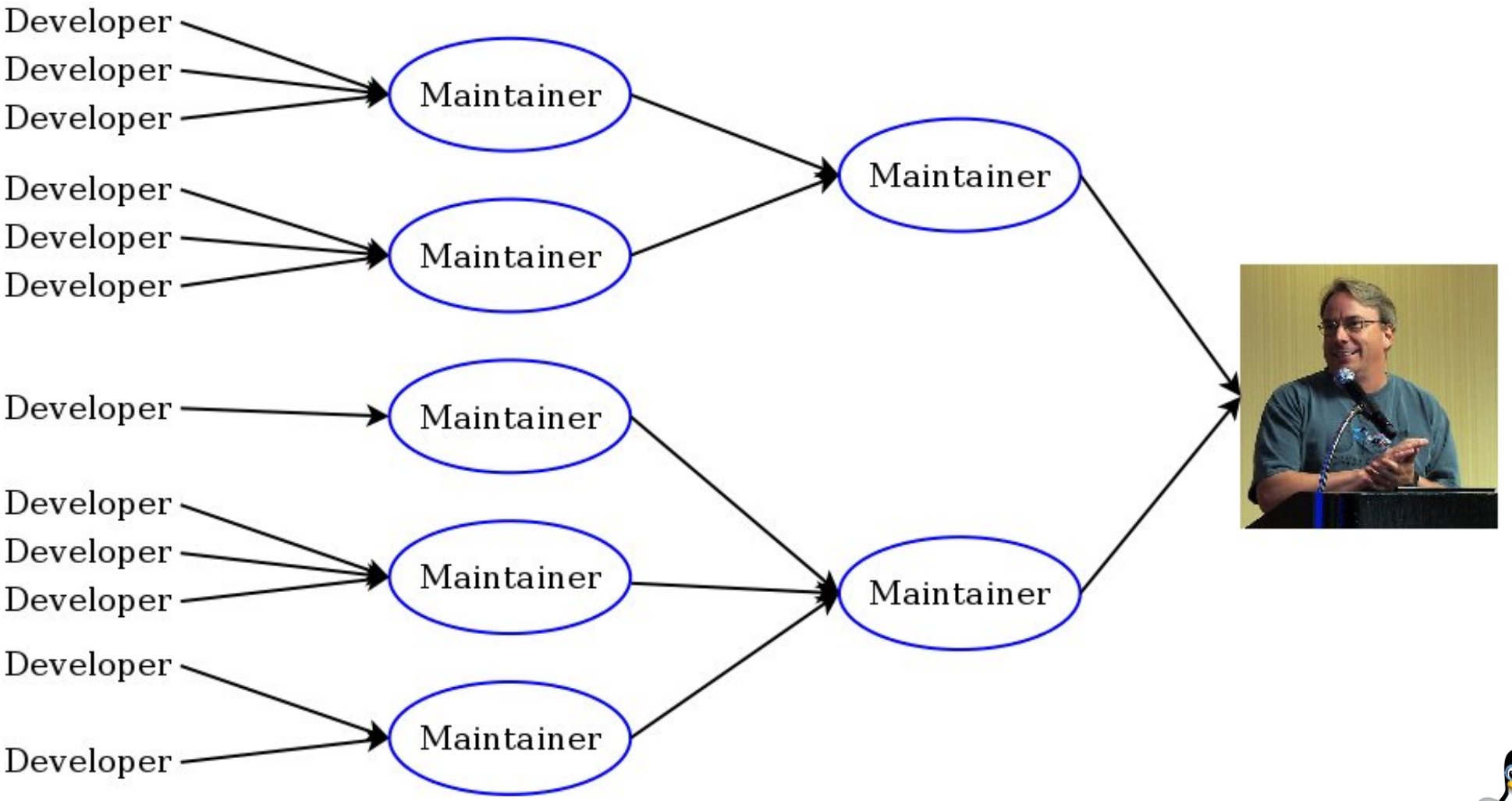
# Interesting kernel facts

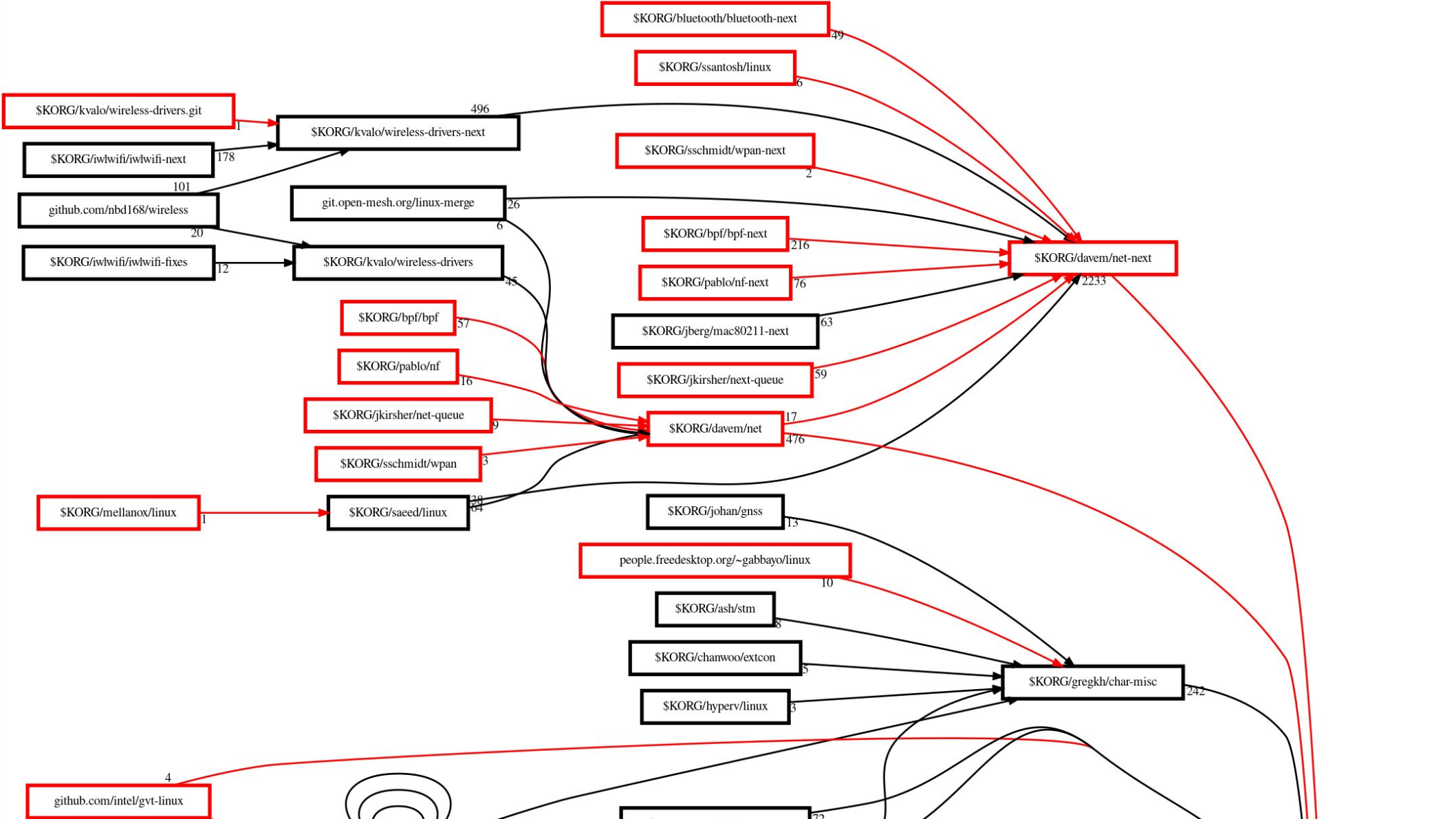
90% (or more) of kernel code is written by paid developers

# Nobody is paid to write kernel documentation

# Interesting kernel facts

The kernel has a well-defined maintainer model


Developer
Developer
Developer
Maintainer
Developer
Developer
Developer
Maintainer
Maintainer
Developer
Maintainer
Developer
Developer
Developer
Maintainer
Maintainer
Developer
Developer
Maintainer


$KORG/bluetooth/bluetooth-next
49
$KORG/ssantosh/linux
6
$KORG/kvalo/wireless-drivers.git
1
496
$KORG/kvalo/wireless-drivers-next
$KORG/iwlwifi/iwlwifi-next
178
$KORG/sschmidt/wpan-next
2
101
github.com/nbd168/wireless
git.open-mesh.org/linux-merge
26
6
$KORG/bpf/bpf-next
216
20
$KORG/iwlwifi/iwlwifi-fixes
12
$KORG/kvalo/wireless-drivers
45
$KORG/davem/net-next
2233
$KORG/pablo/nf-next
76
$KORG/bpf/bpf
57
$KORG/jberg/mac80211-next
63
$KORG/pablo/nf
16
$KORG/jkirsher/next-queue
59
$KORG/jkirsher/net-queue
9
$KORG/davem/net
17
476
$KORG/sschmidt/wpan
3
$KORG/mellanox/linux
1
$KORG/saeed/linux
38
64
$KORG/johan/gnss
13
people.freedesktop.org/~gabbayo/linux
10
$KORG/ash/stm
8
$KORG/chanwoo/extcon
5
$KORG/gregkh/char-misc
242
$KORG/hyperv/linux
3
4
github.com/intel/gvt-linux
72

# The maintainer model

...closely matches the kernel file hierarchy

The SCSI maintainer manages drivers/scsi/

# Documentation does not fit this model

Everybody touches Documentation/

Lots of documentation lives elsewhere

# Kernel developers are conservative

# The end result

Just being the docs maintainer is an interesting challenge

# Kernel documentation in 2016

Over 2,000 .txt files

34 DocBook “template files”

Thousands of kerneldoc comments in source

# 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.
 */
```

# 2016: What’s not to like?

A fragile, complex, home-made build system

# 2016: What’s not to like?

A fragile, complex, home-made build system

No markup in kerneldoc comments

# 2016: What’s not to like?

A fragile, complex, home-made build system

No markup in kerneldoc comments

Ugly formatted output

# 2016: What's not to like?

A fragile, complex, home-made build system

No markup in kerneldoc comments

Ugly formatted output

2,000 standalone bits of text

An unpleasant experience for everybody involved

# What we wanted to do

Preserve readability of plain-text documentation

Easy, plain-text formatting

Create an integrated set of kernel documents

Encourage the creation of more docs

# Something happened in 4.8

## DocBook replaced with Sphinx

- Documentation formatted with RestructuredText

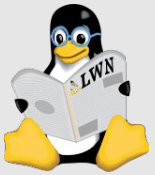
Rebasing
========

"Rebasing" is the process of changing the history of a series of commits within a repository.  There are two different types of operations that are referred to as rebasing since both are done with the ``git rebase`` command, but there are significant differences between them:

- Changing the parent (starting) commit upon which a series of patches is built.  For example, a rebase operation could take a patch set built on the previous kernel release and base it, instead, on the current release.  We'll call this operation "reparenting" in the discussion below.

- Changing the history of a set of patches by fixing (or deleting) broken commits, adding patches, adding tags to commit changelogs, or changing the order in which commits are applied.  In the following text, this type of operation will be referred to as "history modification"

The term "rebasing" will be used to refer to both of the above operations.



# Something happened in 4.8

- DocBook replaced with Sphinx
  - Documentation formatted with RestructuredText
- Kerneldoc comments can use RST

# Something happened in 4.8

- DocBook replaced with Sphinx
  - Documentation formatted with RestructuredText
- Kerneldoc comments can use RST
- Old toolchain thrown away

# The current state of kernel documentation

# In `Documentation/`

3,054 files

(excluding `Documentation/devicetree`)

2,322 in 4.7

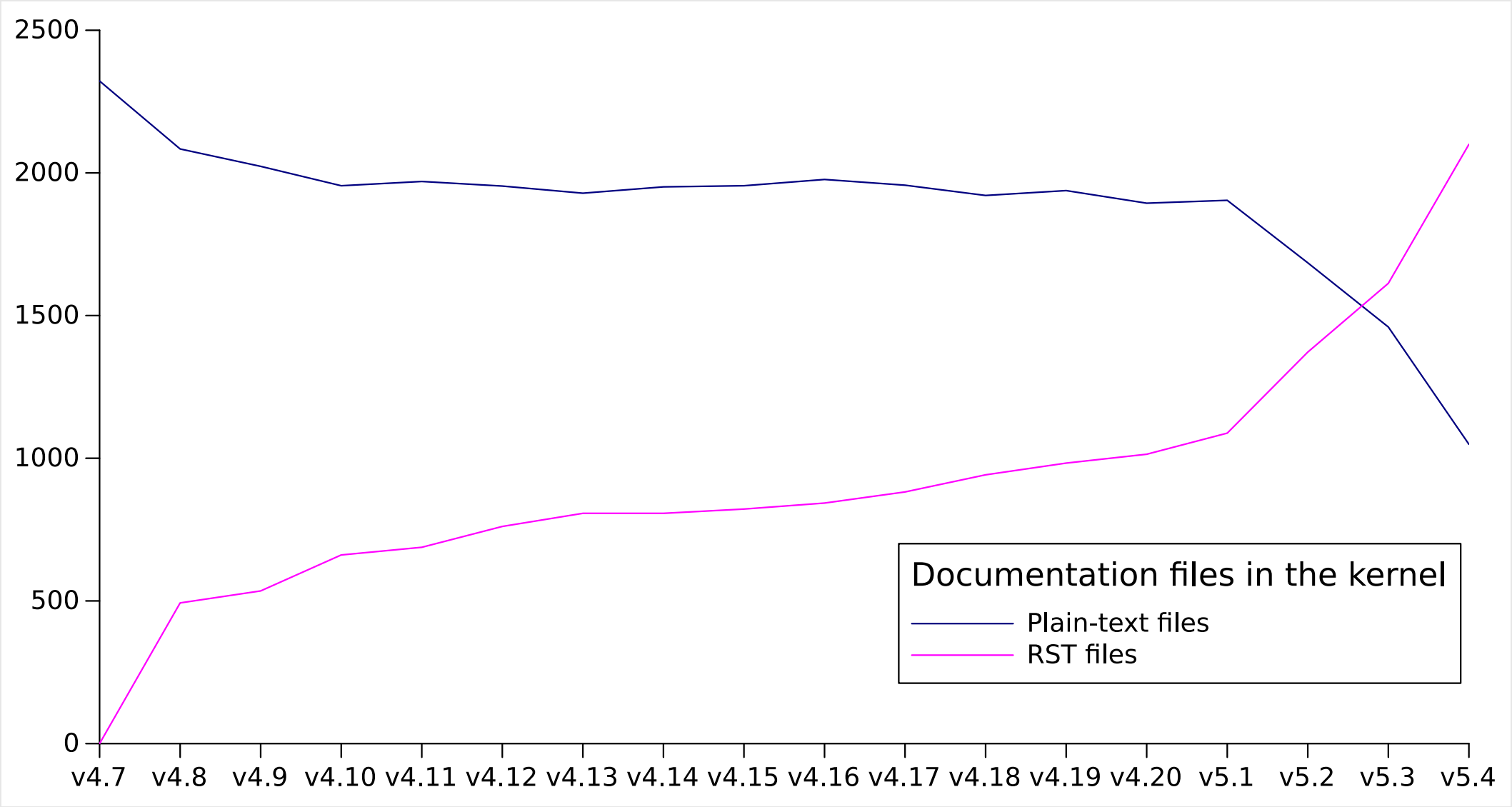
# In `Documentation/`

3,149 files

(excluding `Documentation/devicetree`)

2,322 in 4.7

2,100 `.rst` files


2500
2000
1500
1000
500
0
v4.7
v4.8
v4.9
v4.10
v4.11
v4.12
v4.13
v4.14
v4.15
v4.16
v4.17
v4.18
v4.19
v4.20
v5.1
v5.2
v5.3
v5.4
Documentation files in the kernel
Plain-text files
RST files

# Elsewhere

A vast and growing collection of kerneldoc comments

# 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.
 */
```

# The Linux Kernel API

## List Management Functions

**void list_add(struct list_head * *new*, struct list_head * *head*)**

add a new entry

### Parameters

`struct list_head * new`

new entry to be added

`struct list_head * head`

list head to add it after

### Description

Insert a new entry after the specified head. This is good for implementing stacks.

**void list_add_tail(struct list_head * *new*, struct list_head * *head*)**

add a new entry

### Parameters

`struct list_head * new`

```
/**
 * DOC: dma buf device access
 *
 * For device DMA access to a shared DMA buffer the usual sequence of operations
 * is fairly simple:
 *
 * 1. The exporter defines his exporter instance using
 *    DEFINE_DMA_BUF_EXPORT_INFO() and calls dma_buf_export() to wrap a private
 *    buffer object into a &dma_buf. It then exports that &dma_buf to userspace
 *    as a file descriptor by calling dma_buf_fd().
 *
 * 2. Userspace passes this file-descriptors to all drivers it wants this buffer
 *    to share with: First the filedescriptor is converted to a &dma_buf using
 *    dma_buf_get(). Then the buffer is attached to the device using
 *    dma_buf_attach().
```

```
Basic Operation and Device DMA Access
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

.. kernel-doc:: drivers/dma-buf/dma-buf.c
   :doc: dma buf device access
```

- The DMA buffer FD is also pollable, see Fence Poll Support below for details.

## Basic Operation and Device DMA Access

For device DMA access to a shared DMA buffer the usual sequence of operations is fairly simple:

1. The exporter defines his exporter instance using `DEFINE_DMA_BUF_EXPORT_INFO()` and calls `dma_buf_export()` to wrap a private buffer object into a `dma_buf`. It then exports that `dma_buf` to userspace as a file descriptor by calling `dma_buf_fd()`.
2. Userspace passes this file-descriptors to all drivers it wants this buffer to share with: First the filedescriptor is converted to a `dma_buf` using `dma_buf_get()`. Then the buffer is attached to the device using `dma_buf_attach()`.

   Up to this stage the exporter is still free to migrate or reallocate the backing storage.
3. Once the buffer is attached to all devices userspace can initiate DMA access to the shared buffer. In the kernel this is done by calling `dma_buf_map_attachment()` and `dma_buf_unmap_attachment()`.
4. Once a driver is done with a shared buffer it needs to call `dma_buf_detach()` (after cleaning up any mappings) and then release the reference acquired with dma_buf_get by calling `dma_buf_put()`.

For the detailed semantics exporters are expected to implement see `dma_buf_ops`.

## CPU Access to DMA Buffer Objects

There are mutliple reasons for supporting CPU access to a dma buffer object:

- Fallback operations in the kernel, for example when a device is connected over USB and the kernel needs to shuffle the data around first before sending it away. Cache coherency is handled by braketing any transactions with calls to `dma_buf_begin_cpu_access()` and `dma_buf_end_cpu_access()` access.

  To support dma_buf objects residing in highmem cpu access is page-based using an api similar to kmap. Accessing a dma_buf is done in aligned chunks of PAGE_SIZE size. Before accessing a chunk it needs to be mapped, which returns a pointer in kernel virtual address space. Afterwards the chunk needs to be unmapped again. There is no limit on how often a given chunk can be mapped and unmapped, i.e. the importer does not

## J



## K

# Also...

PDF, EPUB output

Fast incremental builds

scripts/sphinx-pre-install

...

# We have come a long way!

# What’s next?

# Build warnings

If you have expected warnings, you will ignore the new and valid ones. So the only acceptable situation is "no warnings".
— Linus Torvalds, September 26 2019

# Build warnings

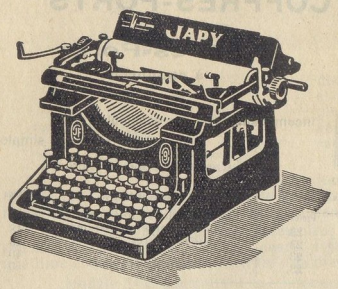
```
./include/linux/netdevice.h:2040: warning:
Function parameter or member 'xps_rxqs_map' not
described in 'net_device'

./include/linux/xarray.h:232: WARNING: Unexpected
indentation.
```

# Convert the remaining .txt files

...in progress...

# Ancient documents






Modèle N° 3

## Machine à Ecrire "JAPY"

**Modèle N° 3**
Ancien modèle reconstruite en usine.
Chariot de 27 m/m.
A partir de 575 frs.

**Modèle N° 9**
Chariot B, 27 m/m 5.
— C, 36 m/m.
— D, 46 m/m 5.
*Prix sur demande.*

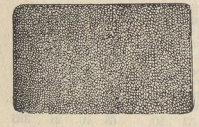
## Machine à Ecrire "JAPY"

**Portative, type V**

*Utilisée dans le Commerce, chez soi et en voyage.*

**Livrée avec coffret gainé et serrure de sûreté.**

*Prix sur demande.*



## Tapis amortisseurs pour Machine à Ecrire

en "CABLOCUIR" avec drap collé au-dessous

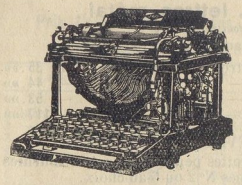
N° 1. Format 34x34 . . . pièce . 16 75
N° 2. — 38x39 . . — . 21.»»

**Gomme PARAPLEX,** *pour nettoyer les caractères de machines à écrire*

Il suffit d'appliquer une ou deux fois fortement la tablette sur les caractères.

N° 807. — La boite de 2 tablettes. . . 5 »»





## Machines à Ecrire d'occasion

Les différentes machines indiquées ci-dessous sont offertes, sauf vente.

*(La liste complète des machines d'occasion est envoyée sur demande).*

Les machines offertes en occasion proviennent d'échanges et de reprises à la vente de machines neuves Elles sont entièrement revisées et remises en état et sont de très bon fonctionnement.

REMINGTON 10, pica . . . . . 950.»»
UNDERWOOD 5, pica . . . . . 1250.»»

**Facilités de paiement —**

## Machines à écrire reconstruites

*Livrées avec la même garantie que les machines neuves.*

UNDERWOOD 5, chariot de 25 m/m . . . . . 1350.»»
UNDERWOOD 3/12, chariot de 30 m/m . . . . . 1450.»»
REMINGTON 12, chariot A de 25 m/m . . . . . 1475.»»
REMINGTON 12, — B de 30 m/m . . . . . 1550.»»

**— Facilités de paiement —**

## Accessoires pour Machines à Ecrire

N° 10. BOITE NÉCESSAIRE contenant burette, flacon d'huile tournevis, brosse dure pour caractères, pinceau de nettoyage, peaussine, gomme ronde, 6 plaques à effacer, la pièce. 36.»»
N° 11. BOITE NÉCESSAIRE (modèle réduit) contenant burette, flacon d'huile, brosse dure pour caractères, pinceau de nettoyage, gomme ronde et 2 plaques à effacer, la pièce. 20.»»
N° 990 Brosse dure pour machine, manche courbé os, . 3.30
N° 993. — soie et laiton. . . . . — . 2.50
N° 995. Brosse douce pour poussière . . . . . — . 2.60
N° 1450. Huile à machine . . . . . le litre . . 20.»»
N° 1451. — — . . . . . le 1/2 litre . . 12.»»
N° 1452. — — . . . . . le 1/4 de litre . 8.»»
N° 1135. Peaussine lavable remplaçant la peau de chamois, 60x35. 6.»»

**TAPIS FEUTRE pour dessous de machine à écrire**
Format 30x30 m/m — Pièce. 9.50

# Ancient documents

**Documentation/platform/x86-laptop-drivers.txt**

```
compal-laptop
=============
List of supported hardware:

by Compal:
    Compal FL90/IFL90
    Compal FL91/IFL91
    Compal FL92/JFL92
    Compal FT00/IFT00

by Dell:
    Dell Vostro 1200
    Dell Mini 9 (Inspiron 910)
    Dell Mini 10 (Inspiron 1010)
    Dell Mini 10v (Inspiron 1011)
    Dell Mini 1012 (Inspiron 1012)
    Dell Inspiron 11z (Inspiron 1110)
    Dell Mini 12 (Inspiron 1210)
```

**From:** Jonathan Neuschäfer <j.neuschaefer@gmx.net>
**To:** linux-doc@vger.kernel.org
**Cc:** linux-arm-kernel@lists.infradead.org, Jonathan Neuschäfer <j.neuschaefer@gmx.net>, Jonathan Corbet <corbet@lwn.net>, Mauro Carvalho Chehab <mchehab+samsung@kernel.org>, linux-kernel@vger.kernel.org
**Subject:** [PATCH 1/2] Documentation/arm/sa1100: Remove some obsolete documentation
**Date:** Thu, 8 Aug 2019 18:58:55 +0200

```
The support for the following boards, among others, was removed in 2004
with commit "[ARM] Remove broken SA1100 machine support.":

- ADS Bitsy
- Brutus
- Freebird
- ADS GraphicsClient Plus
- ADS GraphicsMaster
- Höft & Wessel Webpanel
- Compaq Itsy
- nanoEngine
- Pangolin
- PLEB
- Yopy

Tifon support has been removed in 2.4.3.3.

Signed-off-by: Jonathan Neuschäfer <j.neuschaefer@gmx.net>
---
 Documentation/arm/sa1100/adsbitsy.rst       |  51 ----------
 Documentation/arm/sa1100/brutus.rst         |  69 -------------
 Documentation/arm/sa1100/freebird.rst       |  25 -----
 Documentation/arm/sa1100/graphicsclient.rst | 102 --------------------
 Documentation/arm/sa1100/graphicsmaster.rst |  60 ------------
 Documentation/arm/sa1100/huw_webpanel.rst   |  21 ----
```

# Converting documents to RST?

easy!

Converting documents to RST?

easy!

Evaluating for relevance and correctness?

Updating them to match reality?

...less so

# Organization

Documentation is for the readers

# Who are our readers?

Kernel developers

User-space developers

System administrators

Distributors

End users

...

# Kernel documentation “books”

core-api/
: Core kernel API stuff

userspace-api/
: Stuff for application developers

process/
: How to participate in kernel development

admin-guide/
: Stuff for sysadmins

dev-tools/
: Tools for kernel development

...

# Integration

> Hmmm...probably premature to bring this up, but Documentation/dev-tools/ is kind of thrown together.
> — Brendan Higgins

# Integration

The kernel-doc mechanism is nice, but...

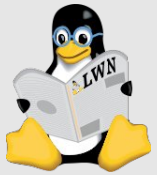
- It does split documents across files

# Missing manuals

Maintainers guide

Subsystem guides for developers

...



# Toolchain improvements

scripts/kernel-doc
- 2200 lines of ancient Perl

PDF generation
- Still depends on LaTeX
- Fragile

Sphinx stylesheets
- ugly!

# Win over doubters

> I don't much care for Documentation/ -- code should be readable and have sufficient comments; I hate rst and I think that anything that detracts from reading code comments in an editor is pure evil.
> — Peter Zijlstra

# Winning over doubters

Sphinx has a syntax for function references:

```
:c:func:`kmalloc()`
```

# Winning over doubters

Sphinx has a syntax for function references:

```
:c:func:`kmalloc()`
```

Automarkup extension added in 5.3

- Just write `kmalloc()` instead

```
 Normal pointers may be stored in the XArray directly.  They must be 4-byte
-aligned, which is true for any pointer returned from :c:func:`kmalloc` and
-:c:func:`alloc_page`.  It isn't true for arbitrary user-space pointers,
+aligned, which is true for any pointer returned from kmalloc() and
+alloc_page().  It isn't true for arbitrary user-space pointers,
 nor for function pointers.  You can store pointers to statically allocated
 objects, as long as those objects have an alignment of at least 4.

 You can also store integers between 0 and ``LONG_MAX`` in the XArray.
-You must first convert it into an entry using :c:func:`xa_mk_value`.
+You must first convert it into an entry using xa_mk_value().
 When you retrieve an entry from the XArray, you can check whether it is
-a value entry by calling :c:func:`xa_is_value`, and convert it back to
-an integer by calling :c:func:`xa_to_value`.
+a value entry by calling xa_is_value(), and convert it back to
+an integer by calling xa_to_value().
```

# Write more documentation!

If you want to be a part of kernel development

...please consider working on documentation

Questions / thoughts?