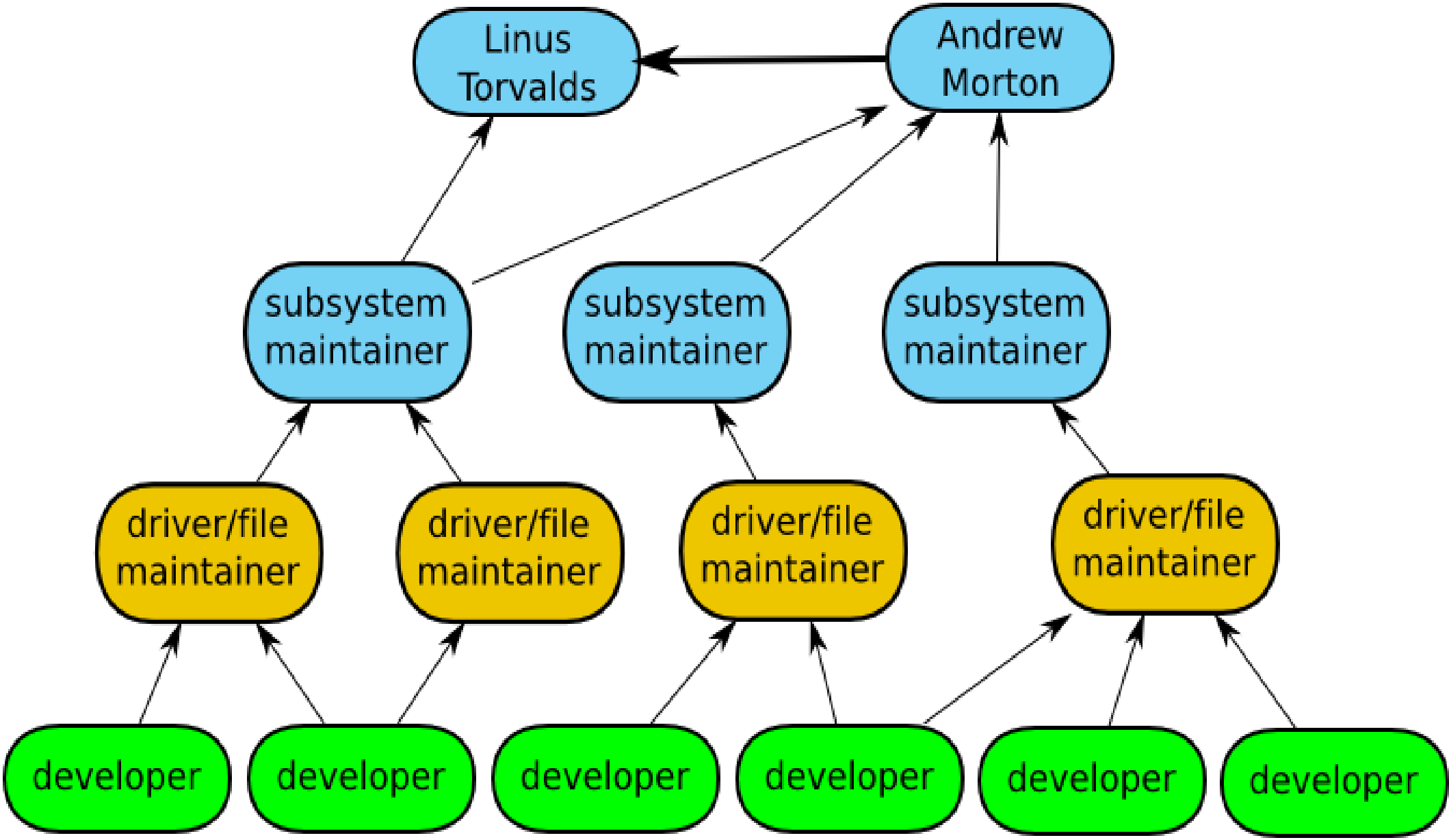
The Linux Kernel Community

How it works, why it matters, and how to get involved

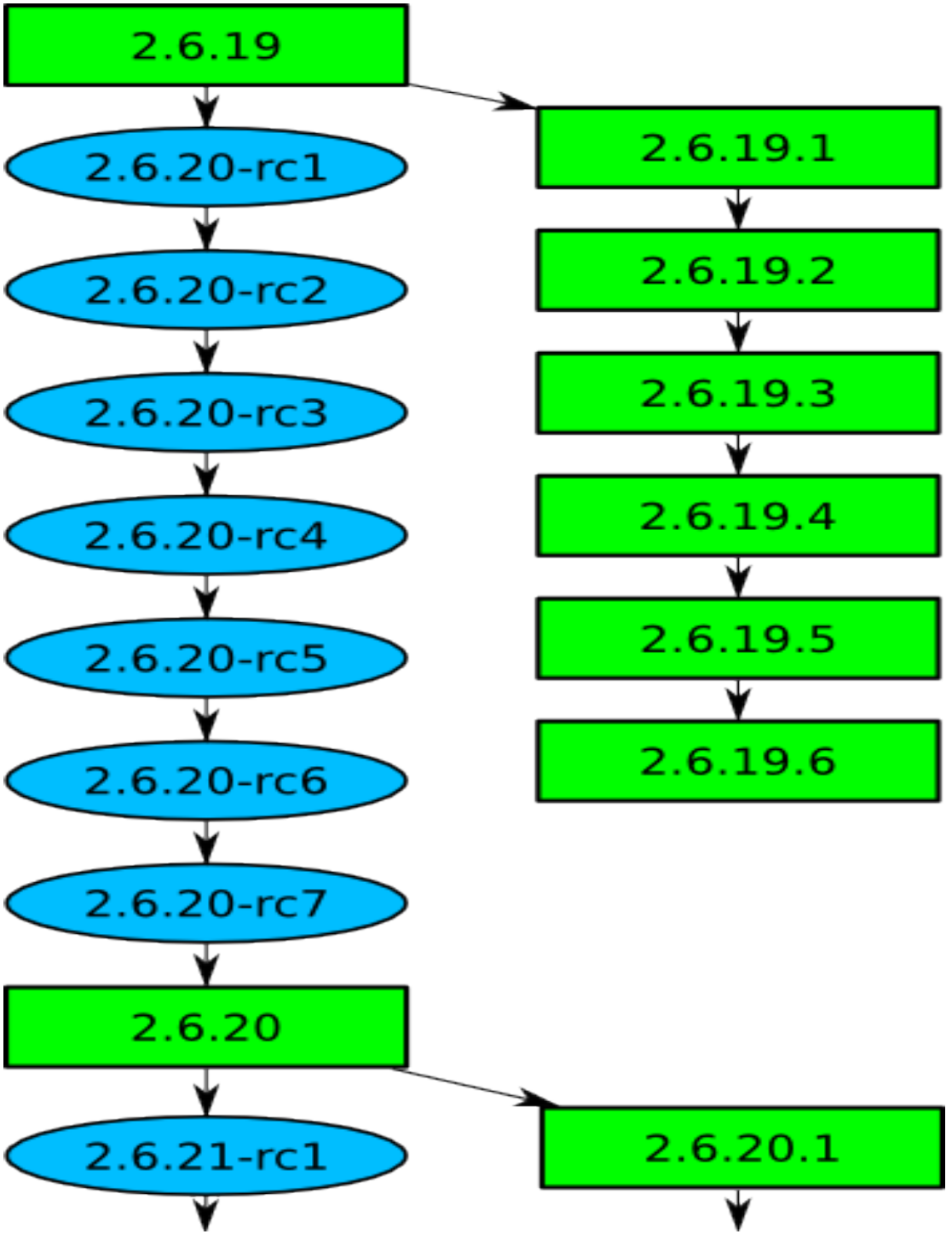
Greg Kroah-Hartman
SuSE Labs / Novell


Linus Torvalds
Andrew Morton
subsystem maintainer
subsystem maintainer
subsystem maintainer
driver/file maintainer
driver/file maintainer
driver/file maintainer
driver/file maintainer
developer
developer
developer
developer
developer
developer

Development or Stable tree?

Development or Stable tree?

Everything is “stable”


2.6.19
2.6.20-rc1
2.6.20-rc2
2.6.20-rc3
2.6.20-rc4
2.6.20-rc5
2.6.20-rc6
2.6.20-rc7
2.6.20
2.6.21-rc1
2.6.19.1
2.6.19.2
2.6.19.3
2.6.19.4
2.6.19.5
2.6.19.6
2.6.20.1


New release every
2½ months

3.31 changes per hour

```
 *
- * Return -EEXIST if there is already a sysfs element with the same name for
- * the same parent.
+ *      LOCKING:
+ *      mutex_lock(parent->i_mutex)
 *
- * Called with parent inode's i_mutex held
+ *      RETURNS:
+ *      Pointer to sysfs_dirent if found, NULL if not.
 */
-int sysfs_dirent_exist(struct sysfs_dirent *parent_sd,
-                       const unsigned char *new)
+struct sysfs_dirent *sysfs_find_dirent(struct sysfs_dirent *parent_sd,
+                                       const unsigned char *name)
 {
-       struct sysfs_dirent * sd;
+       struct sysfs_dirent *sd;

-       for (sd = parent_sd->s_children; sd; sd = sd->s_sibling) {
-               if (sysfs_type(sd)) {
-                       if (strcmp(sd->s_name, new))
-                               continue;
-                       else
-                               return -EEXIST;
-               }
-       }
+       for (sd = parent_sd->s_children; sd; sd = sd->s_sibling)
+               if (sysfs_type(sd) && !strcmp(sd->s_name, name))
+                       return sd;
+       return NULL;
+}

-       return 0;
+/**
```

2.6.18 to 2.6.22

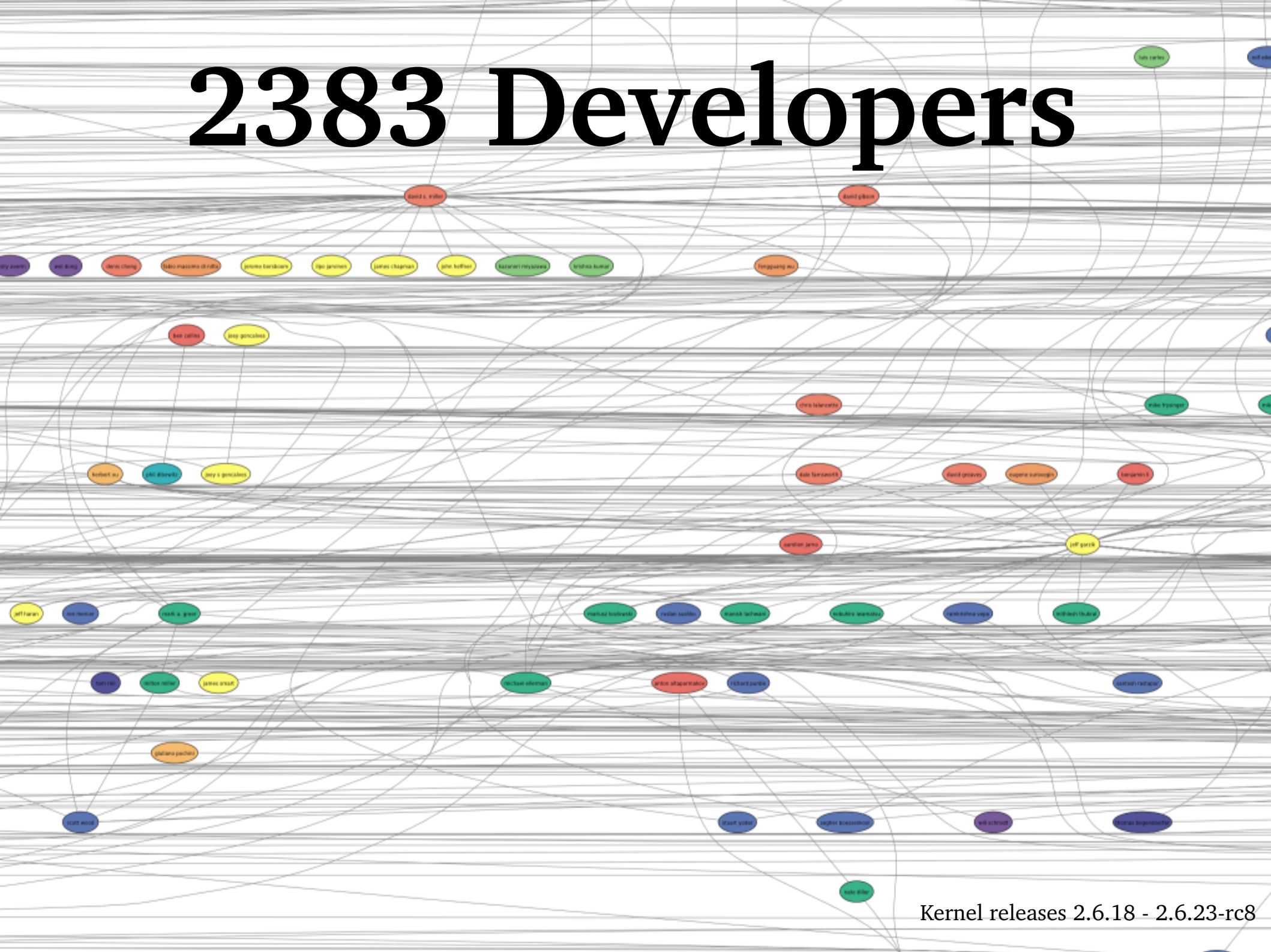
```
if (error)
        goto Error;

/* first, register with generic layer. */
kobject_set_name(&dev->kobj, "%s", dev->bus_id);
error = kobject_add(&dev->kobj);
if (error)
        goto Error;

/* notify platform of device entry */
if (platform_notify)
        platform_notify(dev);

/* notify clients of device entry (new way) */
if (dev->bus)
        blocking_notifier_call_chain(&dev->bus->bus_notifier,
                                     BUS_NOTIFY_ADD_DEVICE, dev);

error = device_create_file(dev, &uevent_attr);
if (error)
        goto attrError;

if (MAJOR(dev->devt)) {
        error = device_create_file(dev, &devt_attr);
        if (error)
                goto ueventattrError;
}

if (dev->class) {
        sysfs_create_link(&dev->kobj, &dev->class->subsys.kobj,
                          "subsystem");
        /* If this is not a "fake" compatible device, then create the
         * symlink from the class to the device. */
        if (dev->kobj.parent != &dev->class->subsys.kobj)
```

8.5 million lines

as of the 2.6.22 release


2330 lines added every day
2800 lines modified every day
24 hours a day
7 days a week
2.6.18 to 2.6.22

2383 Developers



Kernel releases 2.6.18 - 2.6.23-rc8

Top developers by quantity

Al Viro	778
David S. Miller	637
Ralf Baechle	575
Andrew Morton	556
Tejun Heo	537
Adrian Bunk	501
Patrick McHardy	421
Stephen Hemminger	404
Andi Kleen	361
Ingo Molnar	396

Kernel releases 2.6.18 - 2.6.23-rc8

Top Signed-off-by:

Andrew Morton	9130
Linus Torvalds	8291
David S. Miller	3526
Jeff Garzik	2328
Greg Kroah-Hartman	2087
Paul Mackerras	1515
Mauro Carvalho Chehab	1433
Russell King	1175
James Bottomley	1139
Andi Kleen	1125

Kernel releases 2.6.18 - 2.6.23-rc8

Who is funding this work?

1.		
2.		
3.	Red Hat	12.6%
4.	IBM	8.5%
5.	Novell	7.2%
6.	Linux Foundation	6.8%
7.	Intel	4.1%
8.	SGI	1.6%
9.	MIPS	1.5%
10.	Consultants	1.5%

Kernel releases 2.6.18 - 2.6.23-rc8

Who is funding this work?

1. Unknown Individuals	18.1%
2. “Amateurs”	13.3%
3. Red Hat	12.6%
4. IBM	8.5%
5. Novell	7.2%
6. Linux Foundation	6.8%
7. Intel	4.1%
8. SGI	1.6%
9. MIPS	1.5%
10. Consultants	1.5%

Kernel releases 2.6.18 - 2.6.23-rc8

Who is funding this work?

...

139. Motorola 0.0%

Kernel releases 2.6.18 - 2.6.23-rc8

Who is funding this work?

...

15. MontaVista 1.1%

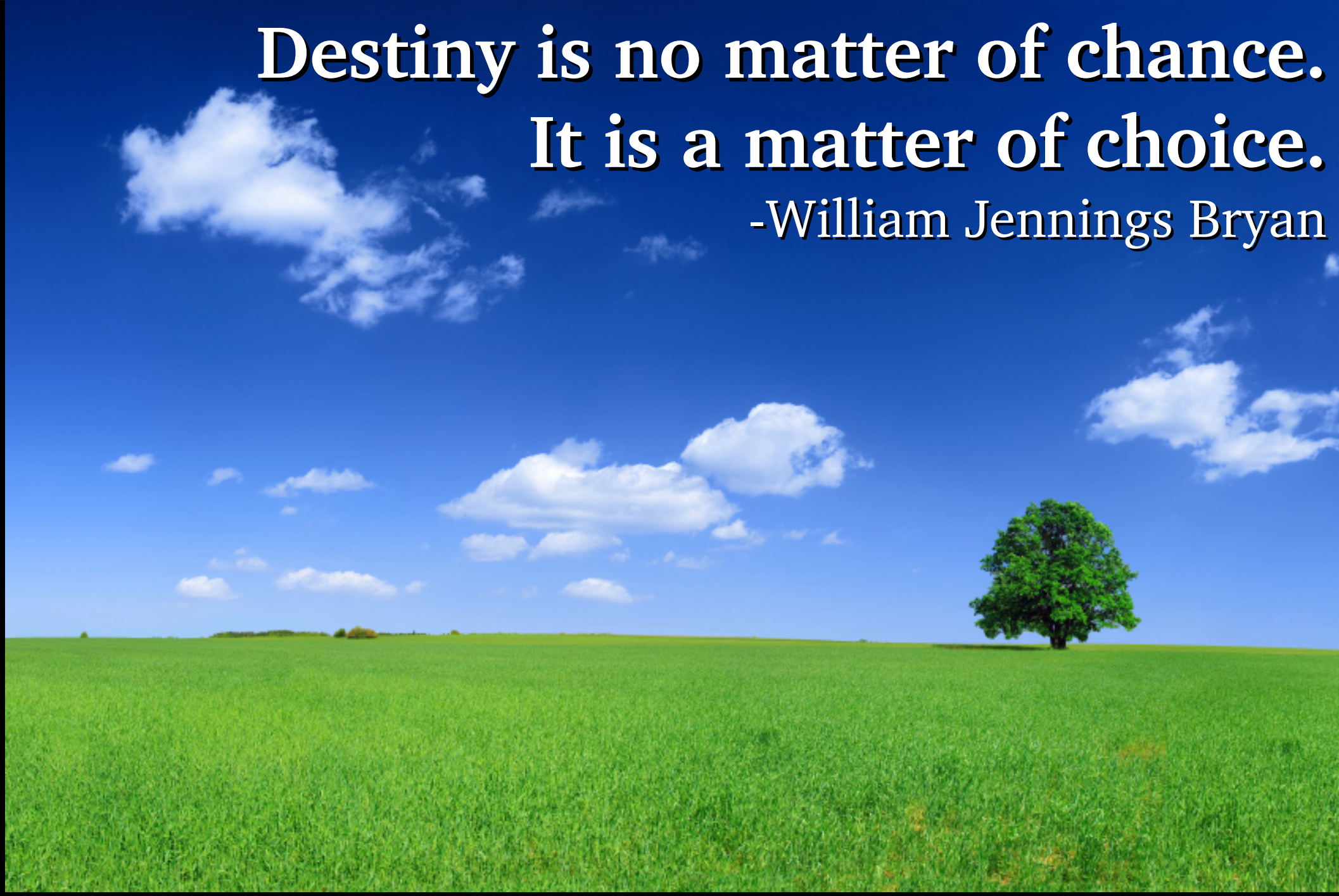
...

139. Motorola 0.0%

Kernel releases 2.6.18 - 2.6.23-rc8

Linux will comprise over 60% of our mobile portfolio in the near future.

- Christy Wyatt


Destiny is no matter of chance.
It is a matter of choice.
-William Jennings Bryan


Documentation/HOWTO

Kernel Newbies


Kernel Janitors

Linux Driver Project

The Linux Driver Project is focused on creating and maintaining OpenSource Linux kernel drivers for all types of devices.

About us

We are a group of Linux kernel developers (over 80 strong) and project managers (over 10) that develop and maintain Linux kernel drivers. We work with the manufacturers of the specific device to specify, develop, submit to the main kernel, and maintain the kernel drivers. We are willing and able to sign NDAs with companies if they wish to keep their specifications closed, as long as we are able to create a proper GPLv2 Linux kernel driver as an end result.

Joining

If you are a company that wishes to have a Linux kernel driver written and maintained by this group, please go to the CompanyProgram page.

www.linuxdriverproject.org

FrequentlyAskedQuestions

We have quite a long list of FrequentlyAskedQuestions that you might wish to peruse before asking them to the developers or on the MailingLists

MailingLists

We currently have two main mailing lists, one for ProjectManagers, and one for Developers. Please see the MailingLists page for more details.

History

This project was started with an announcement to the Linux Kernel mailing list by GregKH, and has enabled many different manufacturers to achieve full Linux support for their devices.

Questions?

If you still have questions, or do not see the answer for what you are looking for, please feel free to contact GregKH.