

The Kernel Report

Linux Foundation Japan Symposium
2007 edition

Jonathan Corbet
LWN.net
corbet@lwn.net

The Plan

- 1) A v e r y b r i e f h i s t o r y o v e r v i e w
- 2) T h e d e v e l o p m e n t p r o c e s s
- 3) G u e s s e s a b o u t t h e f u t u r e

1

H i s t o r y

An extremely rushed history of the Linux kernel

0.01	September, 1991
1.0.0	March, 1994
1.2.0	March, 1995
2.0.0	June, 1996
2.2.0	January, 1999
2.4.0	January, 2001
2.6.0	December, 2003

2.6.14	October 27, 2005
2.6.15	January 2, 2006
2.6.16	March 19, 2006
2.6.17	June 17, 2006
2.6.18	September 19, 2006
2.6.19	November 29, 2006
2.6.20	February 4, 2007
2.6.21	April 21, 2007
2.6.22	July ??, 2007

2

The process

Kernel releases are...

Fast!

New kernels every 2-3 months

Kernel releases are...

Major

Every release has

Major new features

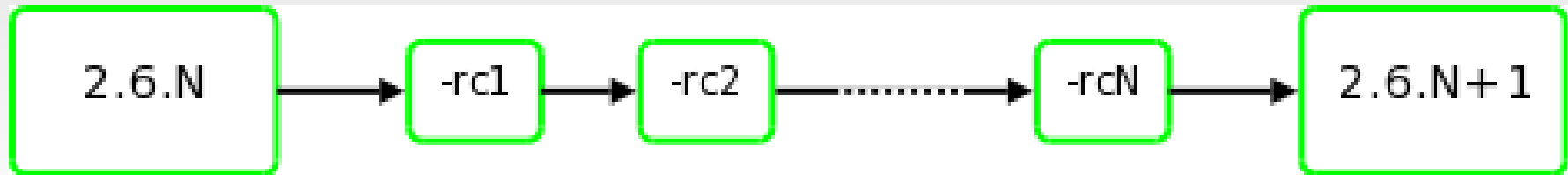
Internal API changes

Kernel releases are...

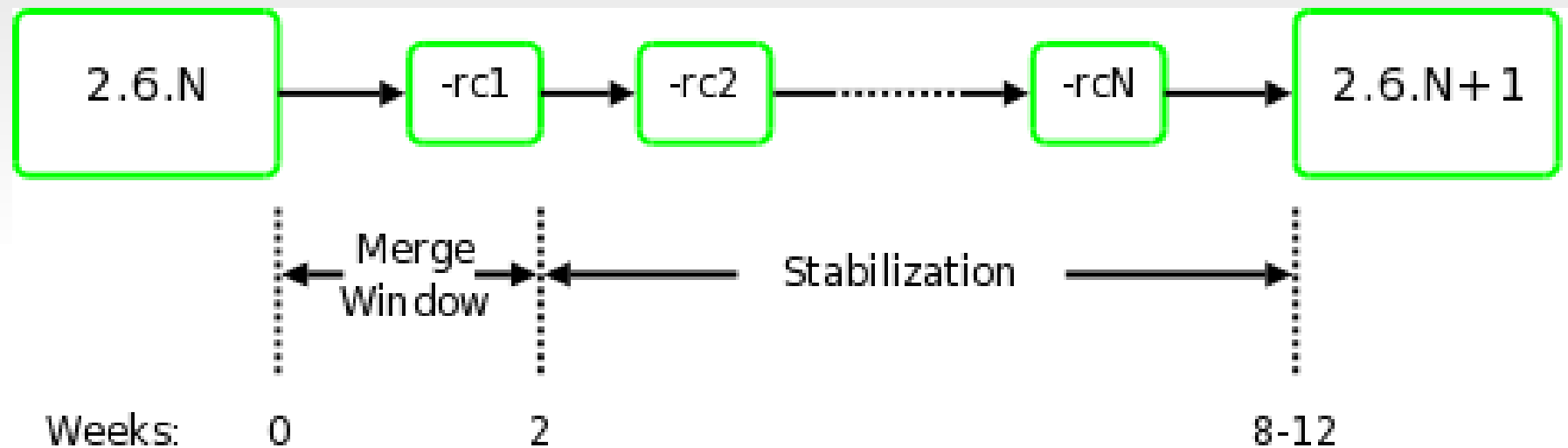
Predictable

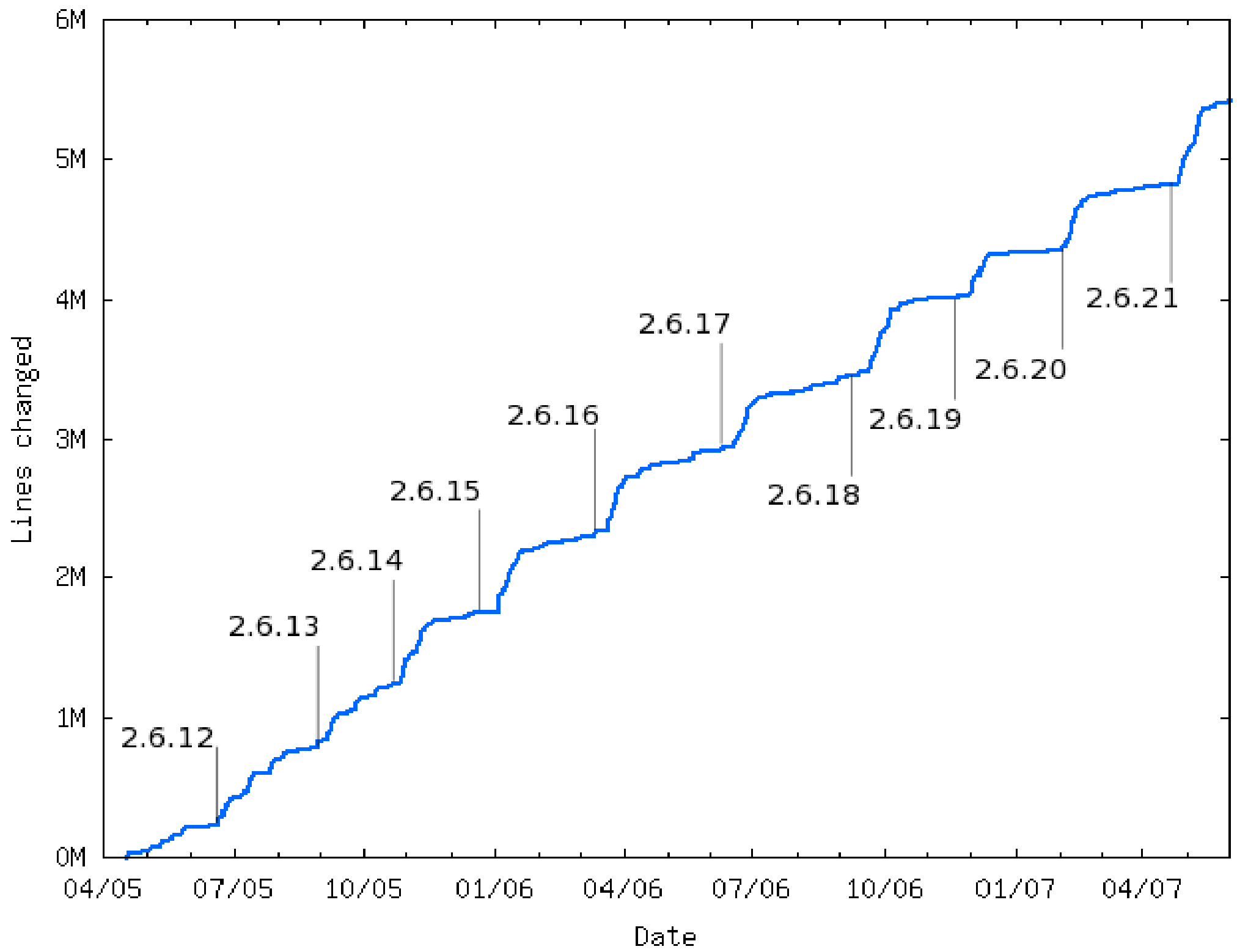
Expect 2.6.23 around October

The release cycle



The release cycle





Who does this work

Since 2.6.17 (June, 2006):

Patches accepted from 2100 developers

30,100 changesets total

Over 2 million lines of code changed

Only 10 contributed $\geq 1\%$ of patches

Largest contributor:

680 changesets

2.3% of the total

Who funds this work

Unknown	27%	S G I	2%
Red Hat	14%	M I P S Tech.	1%
IBM	8%	H P	1%
Novell	7%	C o n s u l t a n t s	1%
Linux Found.	5%	N o k i a	1%
Hobbyists	5%	A s t a r o	1%
Intel	4%	M o n t a V i s t a	1%
Oracle	2%	L i n u x N e t w o r x	1%
Google	2%	Q l o g i c	1%

For more information

→ Greg KH, Thu. 10:00, Emperor

What the process does well

Quickly moves changes to users

What the process does well

Keeps distributors closer to the mainline

What the process does well

Excellent tracking and merging of patches

What doesn't work so well

What doesn't work so well

“The overall quality of 2.6.21 is pretty
horrific.”

— Dave Jones, Fedora kernel maintainer

What doesn't work so well

Jobs nobody wants to do

Bug tracking

Documentation

Fixing really difficult bugs

What's being done

B e t t e r b u g t r a c k i n g

S t a b i l i z a t i o n r e l e a s e s

A u t o m a t e d t e s t i n g

3

W h a t ' s c o m i n g

```
#i n c l u d e <d i s c l a i m e r . h >
```

The next kernel

2.6.22 is due later this month

Features

mac80211 wireless stack

UBI – flash-aware volume management

IVTV video tuner drivers

New CFQ I/O scheduler

New firewire stack

eventfd() system calls

SLUB allocator

Scalability

Scalability

Today's supercomputer is tomorrow's
laptop.

The leading edge

512 processors works well

4096 still needs some work

Highly-contended locks

Per-CPU data structures

SLUB allocator

For more information

- Corey Gough, Thurs 11:00, King
- Martin Bligh, Thurs 14:00, Emperor
- Mel Gorman, Fri 14:00, Emperor
- Peter Zijlstra, Fri 15:00, Emperor
- Christoph Lameter, Sat 10:00, Emp.

The other side of scalability

Small and embedded systems

Reduce data structures

Eliminate unneeded subsystems

Good power management

...

File systems

D i s k s a r e g e t t i n g l a r g e r

B u t n o t f a s t e r

U s e o f f l a s h s t o r a g e i s i n c r e a s i n g

L i n u x f i l e s y s t e m s a r e g e t t i n g o l d

The fsck problem

Whole-disk read time is increasing

-> filesystem check time increasing

Fixing fsck

chunkfs /tilefs

Subdivide filesystems

Only check parts that were in use

→ chunkfs BOF, Fri. 18:00 King

Btrfs

A completely new filesystem

Extent-based

Subvolumes and snapshotting

Checksumming

Online fsck possible

Offline fsck fast by design

ext4

The successor to ext3

Features

48-bit block numbers

Extents

Nanosecond timestamps

Pre-allocation / delayed allocation

Journal checksums

Online defragmentation

→ A. Mathur, Fri 11:00, Rockhopper

Reiser4

The successor to reiserfs

Features

- Improved performance

- Low-level “plugins”

- Filesystem-as-database

Status: stalled

- Needs a new champion

LogFS

A new flash-oriented file system

Features

- On-media directory tree

- “Wandering tree” log structure

- Strong performance

Virtualization

Running independent systems as guests

High-reliability systems

Server consolidation

Security

“Not just a way of extracting money from
venture capitalists”

Xen

C o m m e r c i a l d e v e l o p m e n t

F e a t u r e - r i c h

S h i p p e d b y s o m e d i s t r i b u t o r s

S l o w t o g e t i n t o t h e m a i n l i n e

 M a y c h a n g e i n 2.6.23

Lguest

A.k.a. the “Rustyvisor”

Extremely simple, full virtualization

Probable for 2.6.23

→ Rusty Russell, Fri 10:00, Emperor

KVM

Kernel Virtual Machine

Full virtualization with hardware spt.

Live migration working

Merged in 2.6.20

Still stabilizing post 2.6.22

For more information

General virtualization

- Justin Forbes, Wed 16:00, Emperor
- Jun Nakajima, Fri 12:00, King

KVM

- Avi Kivity, Wed 11:00, Emperor
- Ryan Harper, Wed 15:00, Emperor
- L. Ionkov, Thu 16:00, Emperor

Containers

Lightweight virtualization

All guests share the host kernel

Several projects going

Supporting containers

Containers complicate the code

Indirection for all global resources

Processes

Devices

Filesystems

System time

Resource management needed too

Supporting containers

Multiple container APIs won't fly

Projects must work together

...and they are working together

Some pieces merged now

Some coming soon

Generic process container mechanism

For more info

- Paul Menage, Wed 11:00, King
- H. Pötzi, Wed 12:00, Emperor
- Balbir Singh, Fri 14:00, Fjordland

CPU scheduling

Once it seemed like a solved problem ...

CFS

The Completely Fair Scheduler

Dump complex interactivity heuristics

Use a simpler fairness algorithm

2.6.23

...maybe

Threadlets

A s y n c h r o n o u s s y s t e m c a l l s

I f a s y s t e m c a l l b l o c k s

C o n t i n u e i n a n e w t h r e a d

C o l l e c t t h e r e s u l t s l a t e r

→ Z a c h B r o w n , F r i 1 0 : 0 0 , K i n g

Real time

Hard real time !

Much of this work already merged

Mutexes

Priority inheritance

High-resolution timers

Tickless kernel

Real time - what's left

Sleeping spinlocks

Interrupt handlers in threads

→ Steven Rostedt, Thu 15:00, King

Network channels

...not this year

Wireless networking

The mac80211 stack has been merged
(Formerly Devicescape)

Drivers are a little slower

Various open issues

Regulatory compliance

Wireless drivers

Great support for many adapters

Biggest gap: Atheros

...and it's in the works

Video drivers

The biggest remaining problem area

Intel: supported

Nvidia: not supported

nouveau.freedesktop.org

ATI:

R200, R300 well supported

Reverse-engineered R500 driver out

W h a t a b o u t t h e c l o s e d - s o u r c e d r i v e r s ?

Closed-source drivers

...threaten our free operating system .

Power management

Lots of hardware hassles

Many issues in user space

Things are getting better

- Tickless kernel

- Drivers being fixed

- Power top

→ Len Brown, Thu 15:00, Emperor

Tracing

D trace envy

Tracing

Coming soon: utrace

New in-kernel monitoring / control
2.6.23?

Later:

Kernel markers

Linux trace toolkit

System Tap

→ James Keniston, Wed 16:00, Emperor

Participation

It works best when everybody joins in
Get code into the kernel early

This is hard for some companies

Interfacing with the community is hard

No benefit to mainline inclusion seen

Simple lack of understanding

Continued work toward understanding

GPLv3

It's (almost) official

June 29

Relatively unpopular in kernel circles

The kernel is explicitly GPLv2

A change to GPLv3 would be difficult

Questions ?

slides at <http://lwn.net/talks/ols2007/>

Who oversees this work

Non-author signoffs since 2.6.17:

Linux Foundation	21 %
Google	21 %
Red Hat	16 %
Novell	10 %
unknown	8 %
IBM	6 %
hobbyists	4 %
Intel	4 %
SteelEye	2 %
Cisco	1 %
MIPS Technologies	1 %

Event handling API

Many systems have an event API
not Linux

eventfd gets us closer
2.6.22

kevents remain on the horizon

Low-level support

`paravirt_ops`

The common hypervisor interface

Merged in 2.6.20

Evolving still

`VMI`

VMWare hypervisor interface

2.6.21

Closed-source drivers

...are buggier

Closed-source drivers

...limit our choices

Closed-source drivers

...cannot be supported

Closed-source drivers

...do not give back to the kernel

Closed-source drivers

...can slow kernel development

Closed-source drivers

...are of questionable legality