The Kernel Report

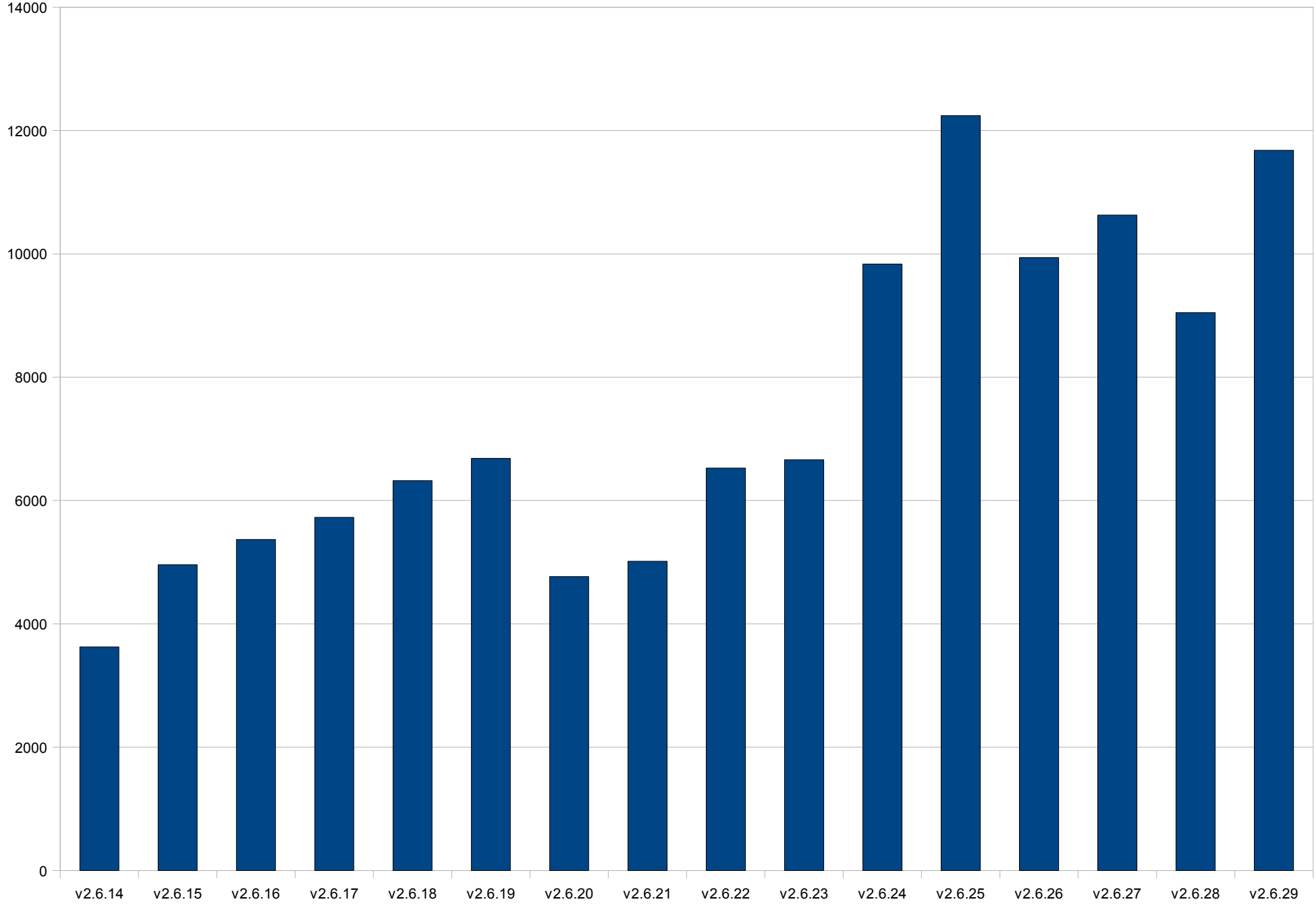
FreedomHEC Taipei '09 edition

Jonathan Corbet
LWN.net
corbet@lwn.net

“Famous last words, but the actual patch volume _has_ to drop off one day. We have to finish this thing one day."

-- Andrew Morton
September, 2005 (2.6.14)

Changesets merged for release



2.6.26 -> 2.6.30

(July 13 2008 to June 5 2009)

43,000 changesets merged
2,300 developers
296 employers

The kernel grew by 2.1 million lines

2.6.26 -> 2.6.30

(July 13 2008 to June 5 2009)

43,000 changesets merged
2,300 developers
296 employers

The kernel grew by 2.1 million lines

In other words:

131 changes per day
6500 lines added every day

The employer stats

None	18%	Analog Devices	1%
Red Hat	12%	academics	1%
unknown	8%	Parallels	1%
Intel	7%	Sun	1%
IBM	6%	Atheros	1%
Novell	6%	AMD	1%
Oracle	4%	Nokia	1%
consultants	3%	Marvell	1%
Fujitsu	2%	SGI	1%
Renesas Tech	1%	Vyatta	1%

2.6.25 (April 16, 2008)

Hardware support
- ath5k driver
- R500 support

ext4 filesystem improvements

SMACK security module

Realtime group scheduling

Memory usage controller

2.6.26 (July 13, 2008)

Read-only bind mounts

More network namespaces

x86 PAT support

KGDB

2.6.27 (October 9, 2008)

Ftrace

UBIFS

Multiqueue networking

gspca video driver set

Block layer integrity checking

2.6.28 (December 24, 2008)

GEM graphics memory manager

ext4 is no longer experimental

-staging tree

Wireless USB

Container freezer

Tracepoints

2.6.29 (March 23, 2009)

Kernel mode setting

Filesystems
- Btrfs
- Squashfs

WIMAX support

4096 CPU support

2.6.30 (June 9?)

TOMOYO Linux

Integrity measurement

ext4 robustness fixes

R6xx/R7xx graphics support

Adaptive spinning mutexes

Object storage device support

FS-Cache

Nilfs

preadv()/pwritev()

Threaded interrupt handlers

...about finished?

...about finished?

...so what's left?

Networking

“Based on all the measurements I'm aware of, Linux has the fastest & most complete stack of any OS.”

-- Van Jacobson

Packet filtering and firewalling

iptables has served us well since 2.4

Problems:

- Much duplicated code
- Difficult user-space interface
- Inflexible

Nftables

Remove protocol-awareness from the kernel
- ...replace with a dumb virtual machine

Rules are translated in user space

Advantages
- Much smaller code base
- Greater flexibility
- Better performance

Other networking stuff

Network namespace development
...still...

Netfilter improvements

Reliable datagram sockets
for 2.6.30

802.15.4 stack (Zigbee and more)

Filesystems

How do we replace ext3/reiserfs/...?

How do we handle solid-state devices?

What guarantees for user space?

ext4

Advantages
- Better performance
- Many limits lifted
- ext3 compatibility

Still stabilizing
- But generally works quite well

Btrfs

A totally new filesystem

Advantages

- Performance
- Full checksumming
- Snapshots
- Internal volume management / RAID

Merged for 2.6.29

- Still very experimental

Others

Nilfs
- Log-structured filesystem
- Versioning/snapshotting
- Merged for 2.6.30

Exofs
- Intended for object storage devices
- Merged for 2.6.30

Network filesystems

CRFS
- Coherent Remote Filesystem
- oss.oracle.com/projects/crfs
- Very early-stage

Pohmelfs
- In -staging for 2.6.30
- Fast filesystem with caching

pNFS
- Distribute NFS across multiple servers
- Linux support in the works

FS-Cache

Local caching for network filesystems
- Big performance boost
- Requires filesystem support

Also useful for slow, local filesystems
- (CDROMs, for example)

Solid-state storage

SSDs present their own challenges
- Transfer size and alignment constraints
- Wear-leveling issues

Enhancements to existing filesystems
- Performance improvements
- Trim support

New filesystems
- UBIFS
- LogFS
- NilFS

Solid-state storage

The longer-term problem:

- SSDs will soon be capable of 100,000+ ops/second
- Will the kernel be able to drive them that fast?

Robustness guarantees

ext3 raised the bar for crash robustness

ext4 tried to lower it again

I want a pony!

“The majority of [application developers] I know felt that ext3 embodied the pony that they'd always dreamed of as a five year old. Stephen gave them that pony almost a decade ago and now you're trying to take it to the glue factory.”

-- Matthew Garrett

What kind of guarantees do we owe our application developers?

New APIs?

fbarrier()

acall()

readdirplus()

reflink()

kevents

A replacement for sockets

“Over the years, we've done lots of nice 'extended functionality' stuff. Nobody ever uses them. The only thing that gets used is the standard stuff that everybody else does too.”

-- Linus Torvalds

Virtualization

Mostly done - in the kernel, at least
- Xen Dom0 still out-of-tree

Remaining work: performance, management

Containers

Lots of namespace work done
- Still stabilizing

Yet to do:
- Resource controllers
- Checkpoint/restart

Photo: photohome_uk

Hardware support

Near universal

A few remaining problems
- Graphics adapters
- Some network adapters

The -staging tree
- A home for substandard drivers

Power management

A variation on the hardware support problem

Power management

Photo: Terren in Virginia

Two approaches

The mainline approach:

- Run each component at the lowest power level

The Android approach:

- Suspend everything whenever possible

Realtime

“While we never had doubts that it would be possible to turn Linux into a real time OS, it was clear from the very beginning that it would be a long way until the last bits and pieces got merged.”

-- Thomas Gleixner

Status of realtime

Code is mostly stable
- Shipped by numerous vendors

User-visible changes are all in mainline

What's not:
- ~~Threaded interrupt handlers~~
- Sleeping mutexes
- Lots of bits and pieces

Security

TOMOYO Linux
- Pathname-based mandatory access control
- 2.6.30

Integrity measurement
- 2.6.30

Still waiting:
- AppArmor
- fanotify

Open issue: sandboxing

Tracing

Photo: Armel Genon

SystemTap

A powerful dynamic tracing environment

Some problems

- Complex, difficult to use
- Requires lots of ancillary data
- Disconnect with kernel community

Alternatives

Ftrace
- Lightweight kernel tracing facility
- Popular with kernel developers

Linux Trace Toolkit
- Well-developed static tracing toolkit
- Extensive user-space tools

But...
- Neither does dynamic tracing
- Neither can trace user-space events

Participation

The kernel development community is growing

We still have trouble with:

- Binary-only modules
- Withheld code
- Language barriers
- Cultural differences
- ...

Documentation/development-process

Questions?