# The kernel report

(LCA 2015 edition)

Jonathan Corbet
LWN.net
corbet@lwn.net

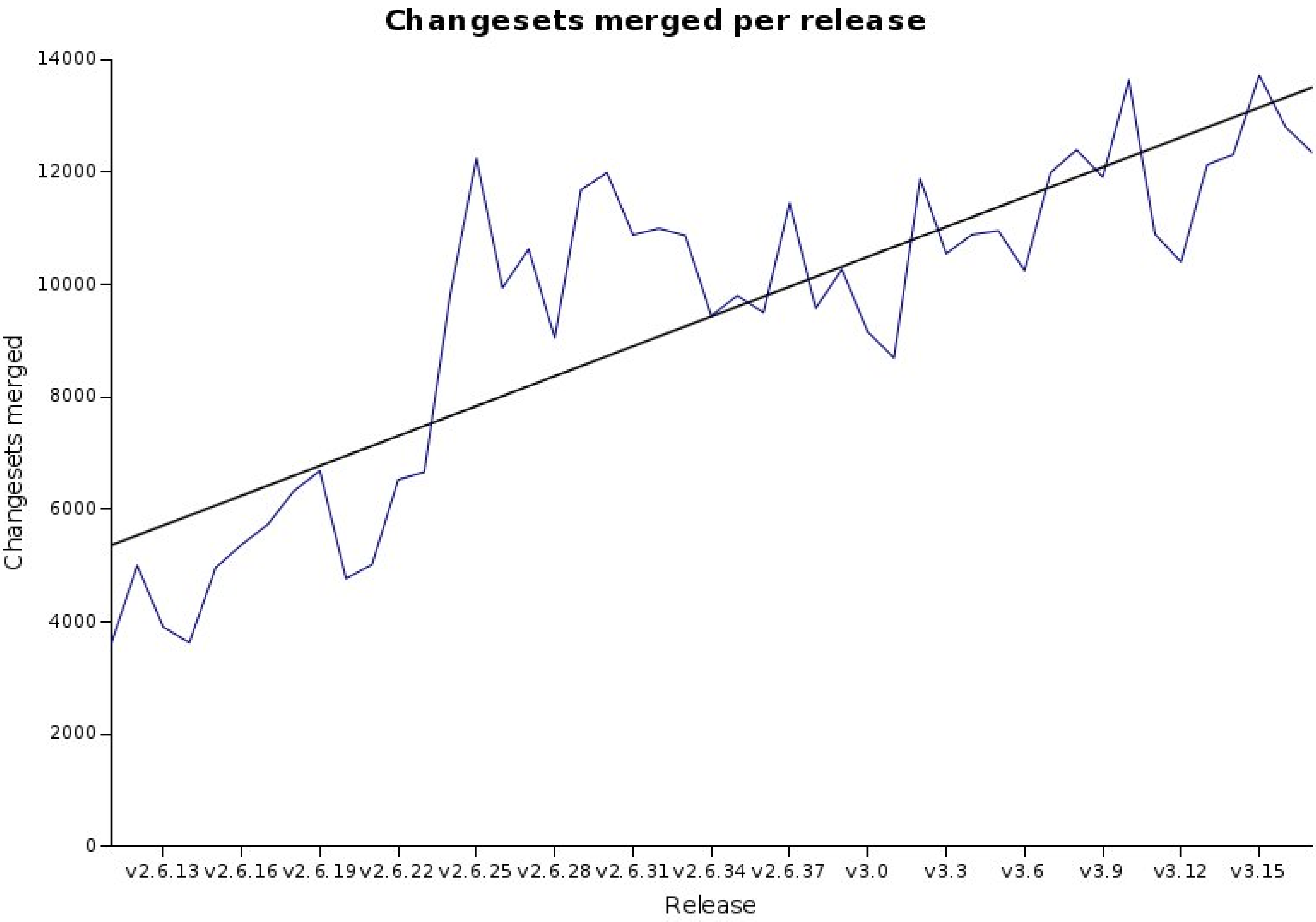
It's nice to be back!

# Recent history

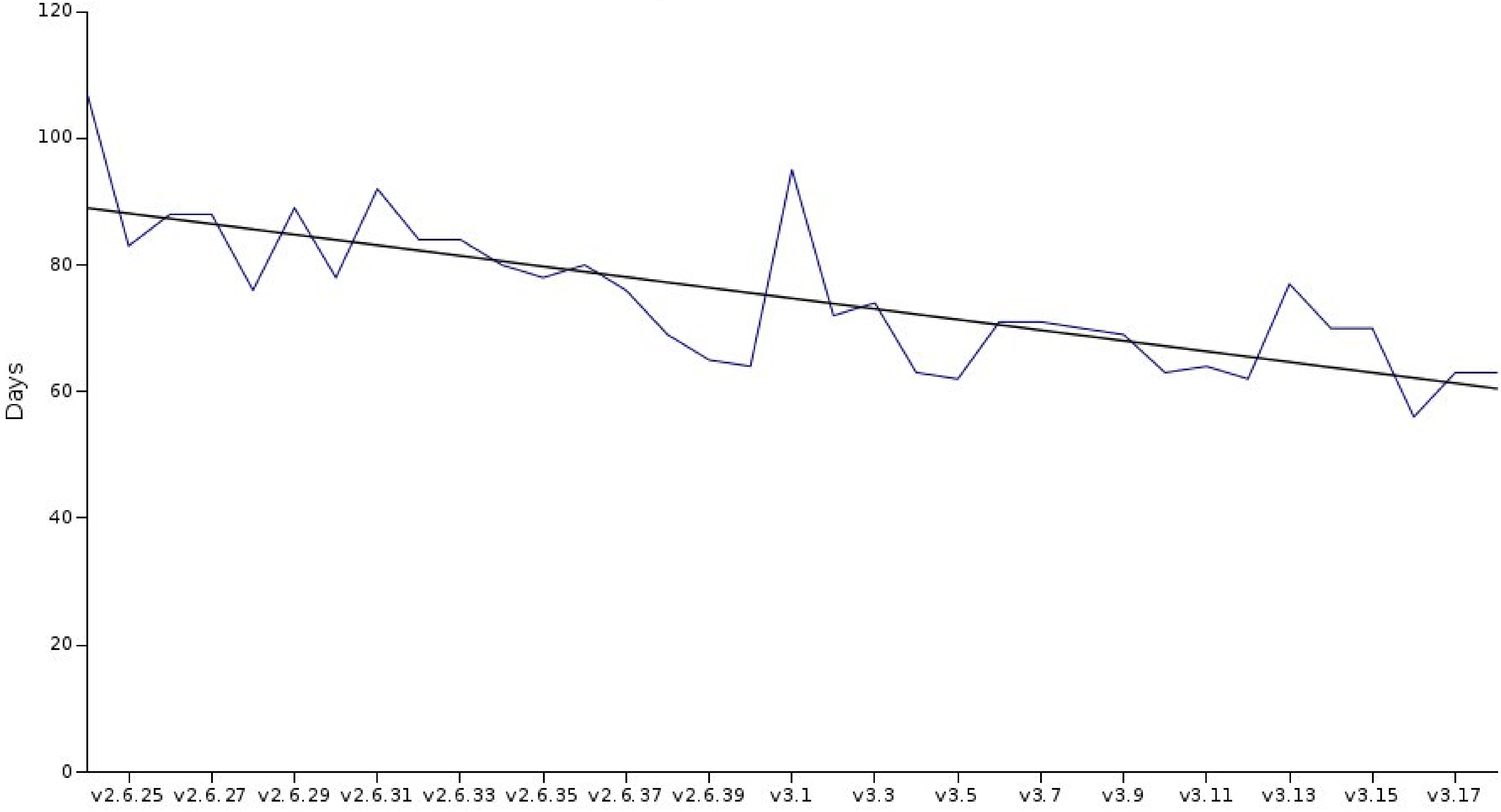
# Recent kernel history

| Vers | Date | Csets | Devs | Days |
|---|---|---|---|---|
| 3.13 | Jan 19 | 12,127 | 1,362 | 77 |
| 3.14 | Mar 20 | 12,311 | 1,306 | 70 |
| 3.15 | Jun 8 | 13,722 | 1,492 | 70 |
| 3.16 | Aug 3 | 12,804 | 1,478 | 56 |
| 3.17 | Oct 5 | 12,354 | 1,433 | 63 |
| 3.18 | Dec 7 | 11,379 | 1,458 | 63 |
| 3.19 | (February) | 11,822* | 1,308* | |

(*so far)


Changesets merged per release
Changesets merged
14000
12000
10000
8000
6000
4000
2000
0
v2.6.13
v2.6.16
v2.6.19
v2.6.22
v2.6.25
v2.6.28
v2.6.31
v2.6.34
v2.6.37
v3.0
v3.3
v3.6
v3.9
v3.12
v3.15
Release

## Development cycle length



# Stable updates

Currently maintained by Greg:

| Vers | Updates | Fixes |
|---|---|---|
| 3.10 | 61 | 3,866 |
| 3.14 | 25 | 2,316 |

# What we've added

Seven new system calls:

```
bpf()
getrandom()
kexec_file_load()
memfd_create()
renameat2()
seccomp()
execveat()
```

# What we've added

Deadline scheduling

Control group reworking

Multiqueue block layer

DRM render nodes

Lots of networking improvements

# ...and, of course...

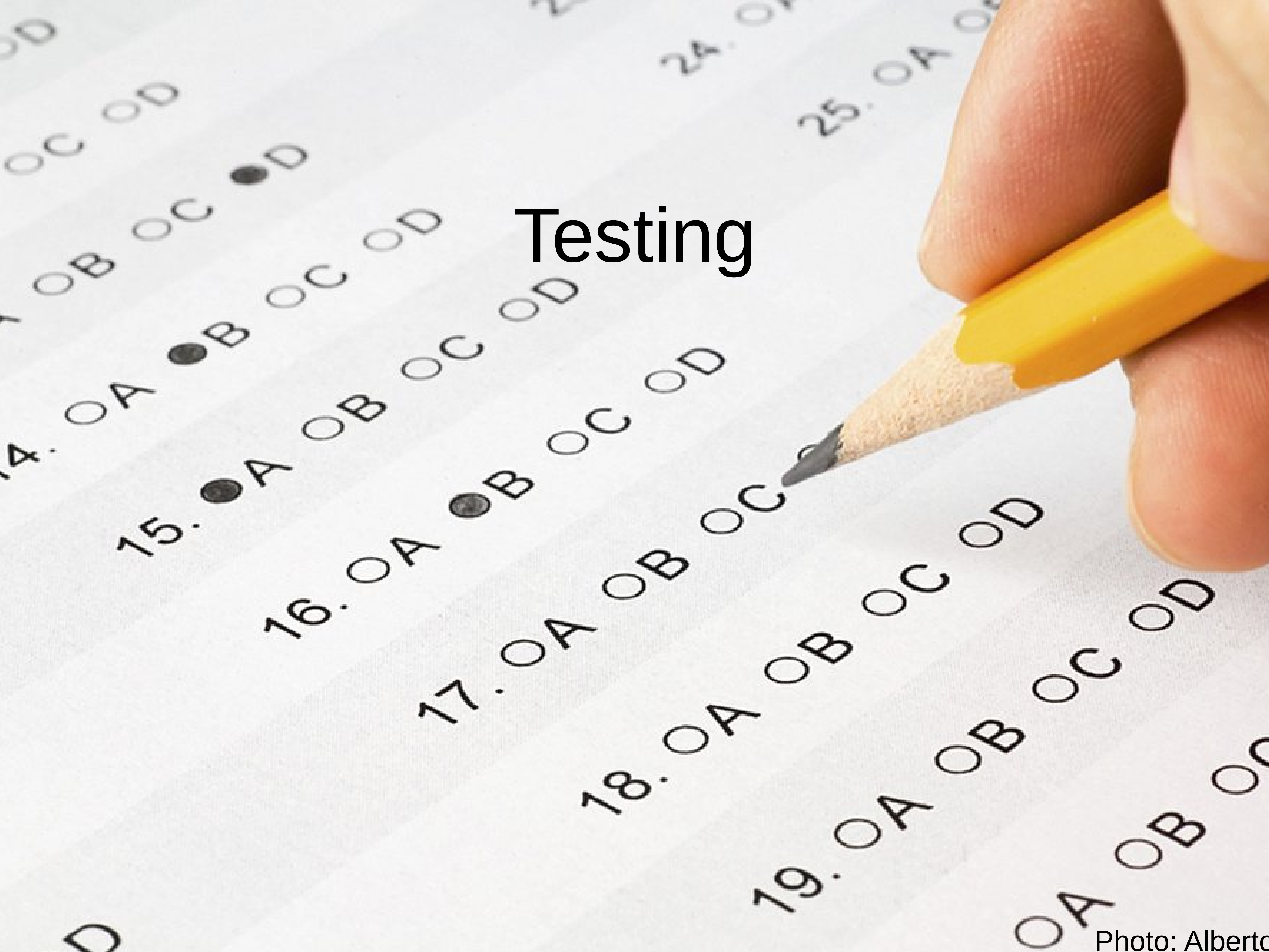
Hundreds of new drivers

# ...and, of course...

Hundreds of new drivers

Thousands of fixes

# A few things I worry about



# Testing

Photo: Alberto

# Better in some ways

linux-next

- Outstanding integration testing

0day build bot

- Immediate feedback on build problems

Coverity, trinity, smatch, Coccinelle, …

- Static analysis, fuzzing, problem highlighting

# Worse in others

# Worse in others

Photo: Samuel Livingston

“Did I just break the kernel”?

# Toward better test frameworks

- A “make test” target for the kernel
  - Rudimentary now, will get better

# Toward better test frameworks

A “make test” target for the kernel

- Rudimentary now, will get better

Encouraging wider-scale testing

- Especially for performance issues

# Performance

# Kernel testing is everybody's business

# Real time

Photo: Jordiet

Real time response in a general-purpose operating system is possible

Real time response in a general-purpose operating system is possible

...if somebody will support the work...

# Security

Photo: stockmonkeys.com

# The bad news

Lots of high-profile security incidents in 2014

115 Kernel CVE's in 2014

Lots of old and unmaintained code

Lots of motivated attackers

Few people working on the problem

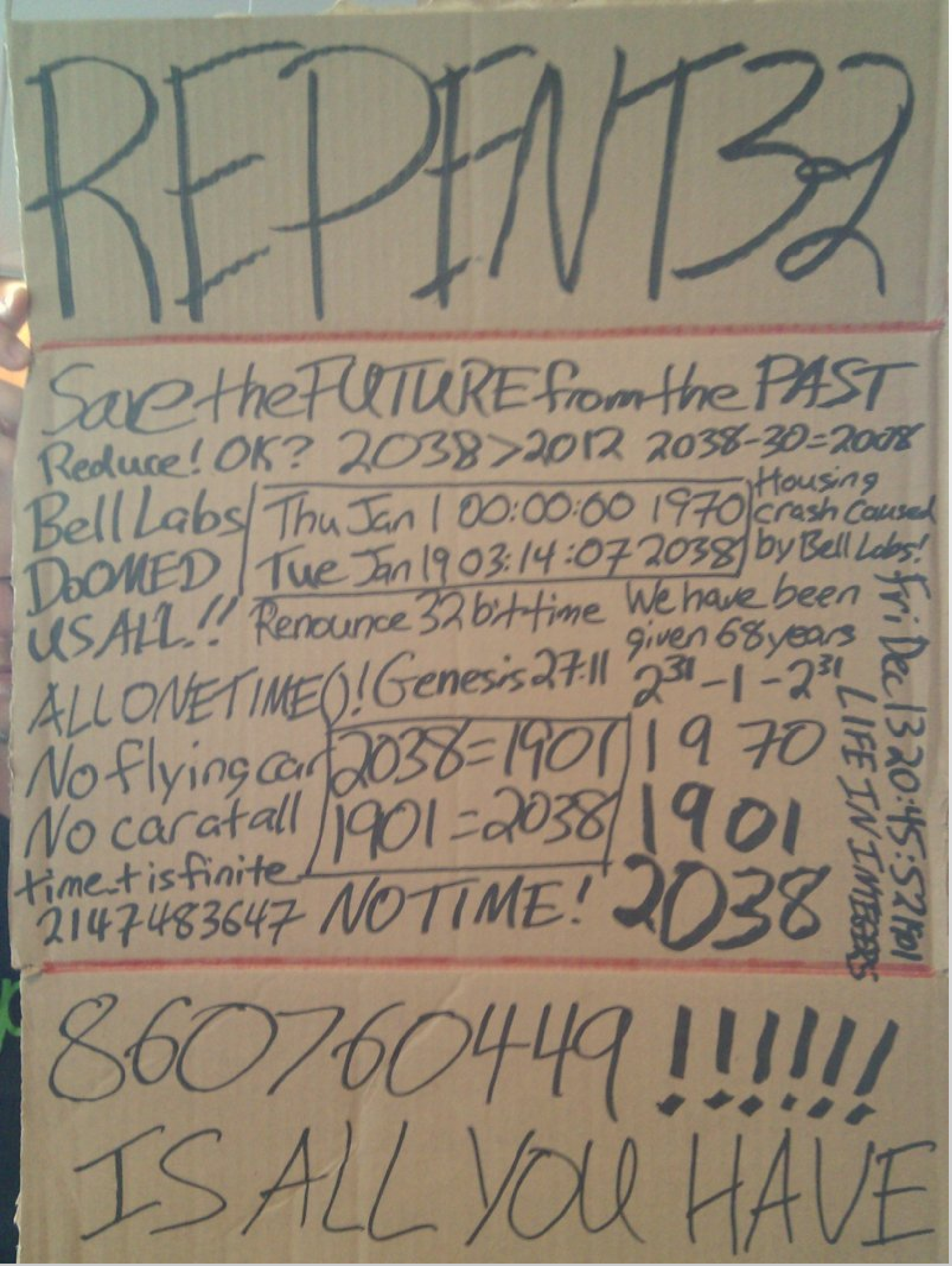
# The goodish news

There were 175 CVEs in 2013

Some effort is going into the problem

- Kernel hardening
- Reducing effects of a compromise

But it's not enough.



2038 is closer than it seems...

Photo: XWRN

# Preparing for 2038

Core timekeeping code: done

# Preparing for 2038

Core timekeeping code: done

New system call APIs: in progress

# Preparing for 2038

Core timekeeping code: done

New system call APIs: in progress

C library preparation: being thought about

# Preparing for 2038

Core timekeeping code: done

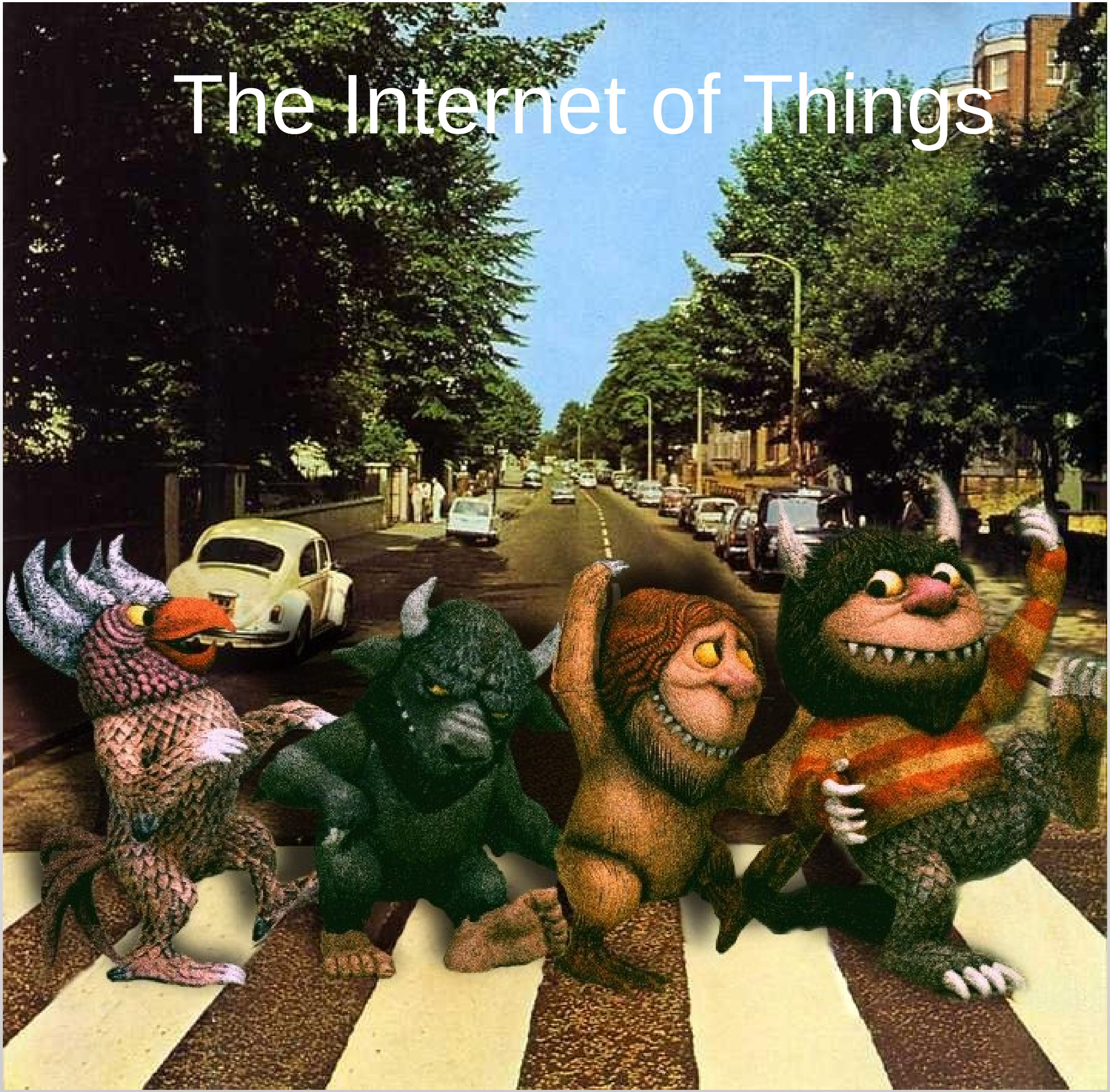
New system call APIs: in progress

C library preparation: being thought about

Fixing applications … don't ask.

# The Internet of Things

# The Internet of Things

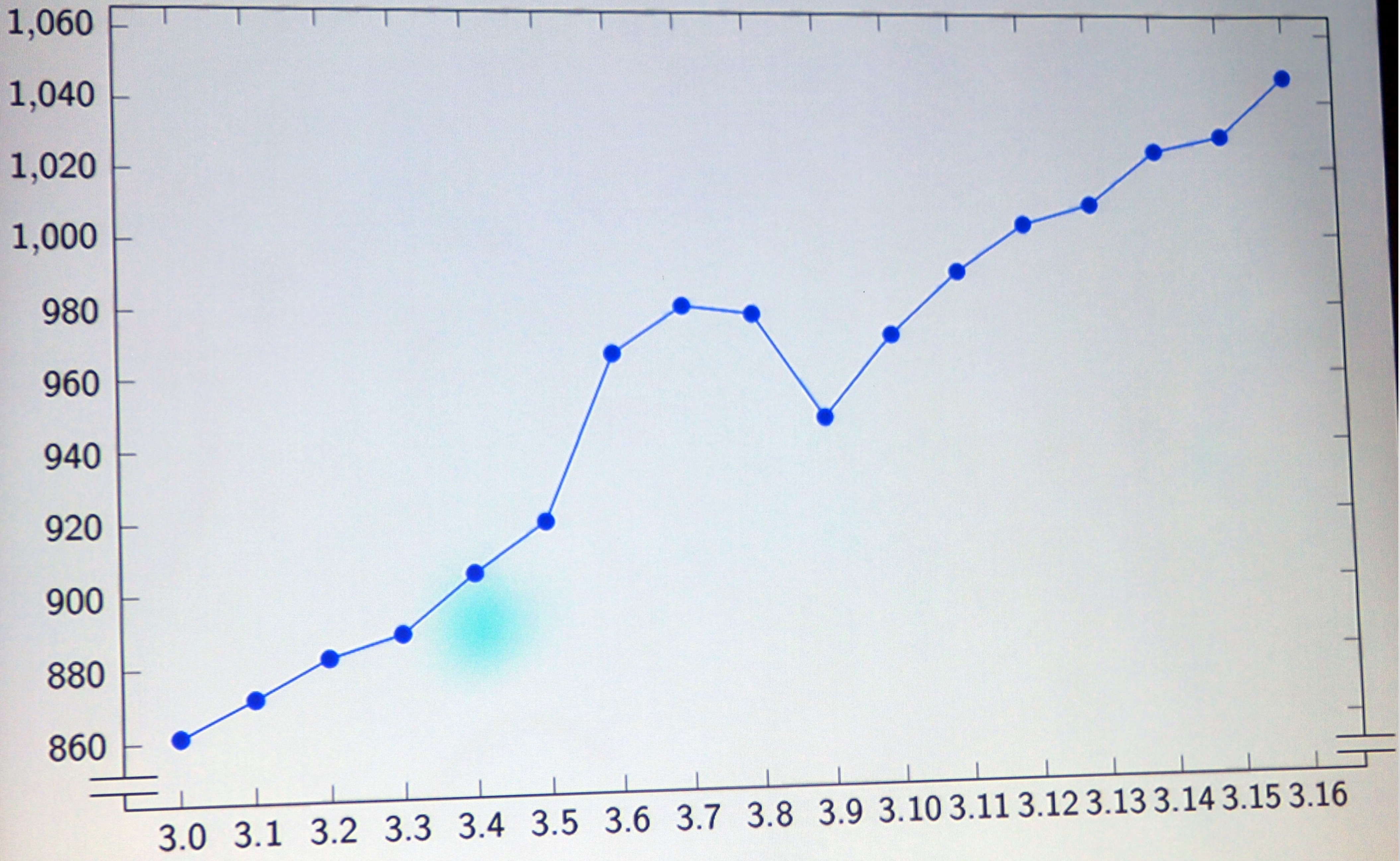


# IoT systems can be small

# IoT systems can be small

...2MB of installed memory, for example...

# minimum kernel size (kB) by kernel version



# Kernel growth will not stop

...we need the features...

# What's to do?

The kernel tinification effort

http://tiny.wiki.kernel.org/

# Tinification challenges

Avoiding a configuration mess

Support

Keeping ahead of growth

Either Linux will be suitable for IoT applications...

Either Linux will be suitable for IoT applications...

...or something else will come along

# New and interesting stuff

# sealed files and memfds

What is a sealed file?

# Not this kind of seal!

Photo: Liam Quinn

# sealed files and memfds

- What is a sealed file?
    - A memory-mapped file whose contents are immutable
    - shmfs only

# sealed files and memfds

What is a sealed file?

- A memory-mapped file whose contents are immutable
- shmfs only

memfd: a sharable, sealable memory area

# sealed files and memfds

What is a sealed file?

- A memory-mapped file whose contents are immutable
- shmfs only

memfd: a sharable, sealable memory area

Result: sharable, unchangeable memory areas

- Merged for 3.17

# kdbus

D-bus-like IPC in the kernel

Why?

- Performance
- Security
- Early availability

Merge probable in 2015

# Virtual machines

Virtual machines in the kernel???

# Virtual machines

Virtual machines in the kernel???

We have:

- ACPI
- Netfilter
- nftables
- tracing filters
- socket filters with BPF
- …

# BPF

“Berkeley Packet Filter”

Originally designed for tcpdump-like tools

Used to filter packets delivered to sockets

- Also with seccomp

# Extended BPF (eBPF)

- More registers (BPF has two)
- New instructions
  - Similar to hardware operations
- Ability to call kernel functions
- Program verifier

- eBPF maps
  - Arrays to share data with the kernel or user space

- Moved out of the networking stack in 3.17

# The future of eBPF

Seccomp filters
Tracing filters
nftables?

… eBPF is becoming the standard kernel VM

# Page fault handling in user space

# Page fault handling in user space

Why???

# Page fault handling in user space

Why?  Virtual machine migration

# Page fault handling in user space

Mark a region for user-space handling:

```
madvise(...MADV_USERFAULT);
```

Get fault notifications with:

```
userfaultfd();
```

Resolve faults with:

```
remap_anon_pages(...);
```

# Live kernel patching

a.k.a. reboots are a pain

# Live kernel patching

We do not lack for options

- KernelCare
- ksplice
- kPatch
- kGraft
- Parallels live patching

# Live kernel patching

We do not lack for options

- ~~KernelCare~~
- ksplice
- kPatch
- kGraft
- Parallels live patching

# Live kernel patching

- We do not lack for options
  - ~~KernelCare~~
  - ~~ksplice~~
  - kPatch
  - kGraft
  - Parallels live patching

# Live kernel patching

We do not lack for options

- ~~KernelCare~~
- ~~ksplice~~
- kPatch
- kGraft
- ~~Parallels live patching~~

# kPatch and kGraft

Both use ftrace machinery

- Catch calls to changed functions
- Divert to a new version

They differ in other ways

# kPatch and kGraft

Both use ftrace machinery

- Catch calls to changed functions
- Divert to a new version

They differ in other ways

Will both be merged?  No way.

# The future of live patching

kGraft and kPatch have agreed on a base layer

Expected to merge for 3.20

# The trouble with crazy new stuff

People use it!

# The trouble with crazy new stuff

People use it!

These features must be supported forever

- ...as must the API

We're not always all that good at designing APIs

- control groups

How can we blaze new trails without making a huge mess of the kernel?

Thank you