The Kernel Report

(LF Japan Symposium 2008 edition)

Jonathan Corbet
LWN.net
corbet@lwn.net

Theme

Challenges / Responses

Challenge

Get the next release out

Response

The 2.6.x release cycle

4-5 releases per year
Each a major release

2.6.24 – January 24, 2008

Network namespaces

Control groups

i386/x86_64 architecture merger

Kernel markers

2.6.25 – April 16, 2008

ath5k wireless driver

Video driver updates (R500)

Realtime group scheduling

ext4 filesystem improvements

memory usage controller

SMACK security module

2.6.26 – July 13, 2008

x86 PAT support

Read-only bind mounts

More network namespace work

KGDB

2.6.27 – October 9, 2008

Block layer data integrity checking

Ftrace

gspca video camera drivers

UBIFS

Multiqueue networking

System call extensions – new flags

2.6.28 – January

ext4dev becomes ext4

Wireless regulatory compliance layer

Lots of block layer work

UWB / Wireless USB support

i915 Graphics Execution Manager

Container freezer

Many tracing improvements

Challenge

Sustain a high rate of development
One of the fastest anywhere

A single kernel cycle involves
10,000+ individual changesets
1,000 developers
1-200 corporations

A single kernel cycle involves
10,000+ individual changesets
1,000 developers
1-200 corporations

2.6.27:
10,600 changesets
1109 developers
150 companies

linux-next

Contains patches for 2.6.n+1
Find integration problems
Early testing

The new development kernel
...sort of

Challenge

Maintaining kernel quality

Too many features, too few fixes?

Responses

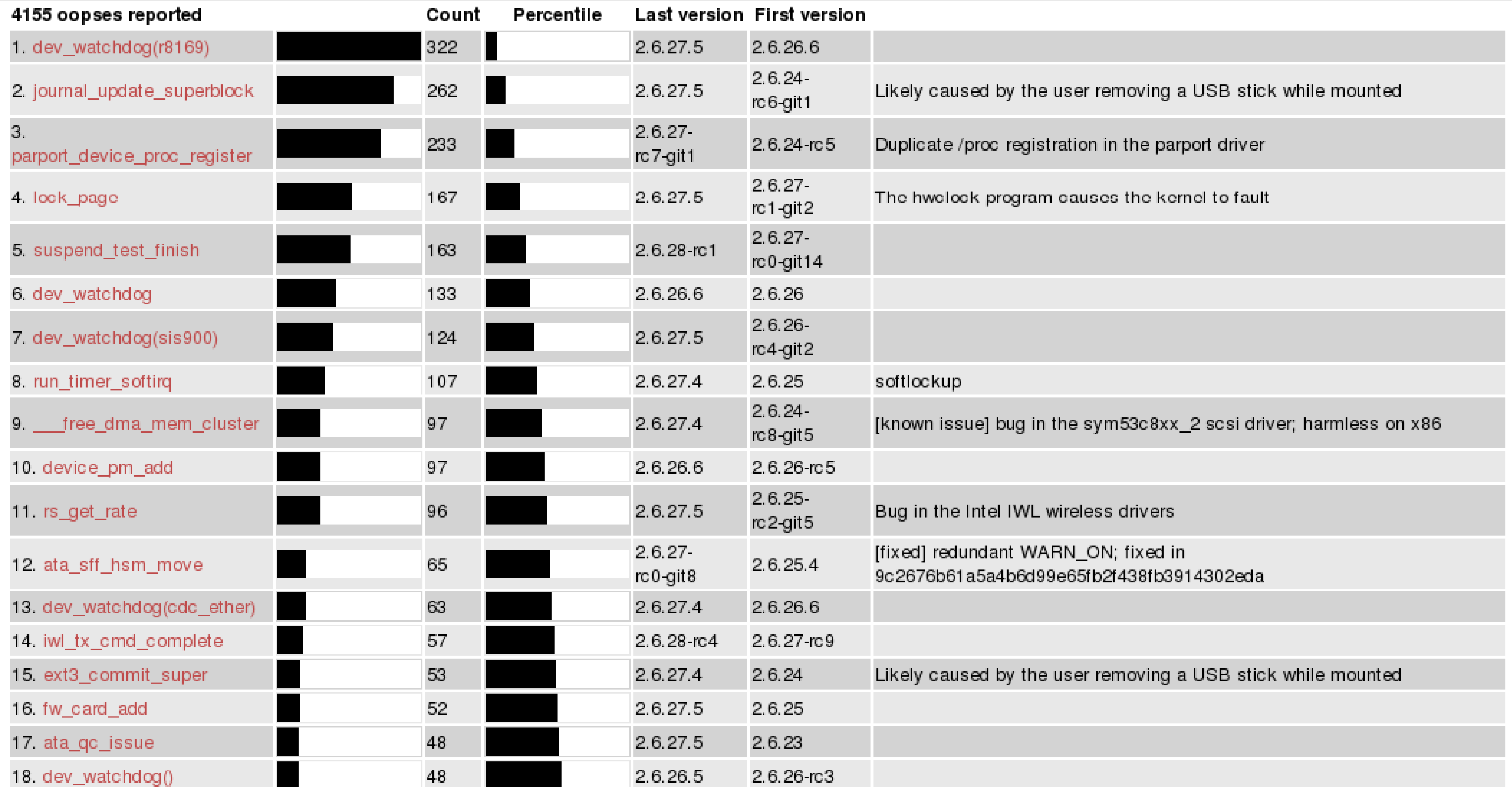
Tracking and fixing of regressions

```
Listed regressions statistics:

  Date             Total  Pending  Unresolved
  ----------------------------------------
  2008-09-12         163       51          38
  2008-09-07         150       43          33
  2008-08-30         135       48          36
  2008-08-23         122       48          40
  2008-08-16         103       47          37
  2008-08-10          80       52          31
  2008-08-02          47       31          20
```

Responses

Better tools



4155 oopses reported		Count	Percentile	Last version	First version	
1. dev_watchdog(r8169)		322		2.6.27.5	2.6.26.6	
2. journal_update_superblock		262		2.6.27.5	2.6.24-rc6-git1	Likely caused by the user removing a USB stick while mounted
3. parport_device_proc_register		233		2.6.27-rc7-git1	2.6.24-rc5	Duplicate /proc registration in the parport driver
4. lock_page		167		2.6.27.5	2.6.27-rc1-git2	The hwclock program causes the kernel to fault
5. suspend_test_finish		163		2.6.28-rc1	2.6.27-rc0-git14	
6. dev_watchdog		133		2.6.26.6	2.6.26	
7. dev_watchdog(sis900)		124		2.6.27.5	2.6.26-rc4-git2	
8. run_timer_softirq		107		2.6.27.4	2.6.25	softlockup
9. __free_dma_mem_cluster		97		2.6.27.4	2.6.24-rc8-git5	[known issue] bug in the sym53c8xx_2 scsi driver; harmless on x86
10. device_pm_add		97		2.6.26.6	2.6.26-rc5	
11. rs_get_rate		96		2.6.27.5	2.6.25-rc2-git5	Bug in the Intel IWL wireless drivers
12. ata_sff_hsm_move		65		2.6.27-rc0-git8	2.6.25.4	[fixed] redundant WARN_ON; fixed in 9c2676b61a5a4b6d99e65fb2f438fb3914302eda
13. dev_watchdog(cdc_ether)		63		2.6.27.4	2.6.26.6	
14. iwl_tx_cmd_complete		57		2.6.28-rc4	2.6.27-rc9	
15. ext3_commit_super		53		2.6.27.4	2.6.24	Likely caused by the user removing a USB stick while mounted
16. fw_card_add		52		2.6.27.5	2.6.25	
17. ata_qc_issue		48		2.6.27.5	2.6.23	
18. dev_watchdog()		48		2.6.26.5	2.6.26-rc3	

Responses

Social pressure + tighter rules

“Here's a simple rule of thumb:
if it's not on the regression list
if it's not a reported security hole
if it's not on the reported oopses list
then why are people sending it to me?”

-- Linus Torvalds

Challenge

The kernel is a common resource

...driven by divergent interests

Response

The “upstream first” policy

No differentiation at the kernel level

Who contributes

2.6.23 -> 2.6.27

(None)	19%	Movial	2%
Red Hat	12%	SGI	1%
IBM	7%	academia	1%
unknown	6%	Analog Devices	1%
Novell	6%	Renasas Tech	1%
Intel	5%	Freescale	1%
Parallels	2%	MontaVista	1%
Oracle	2%	Fujitsu	1%
linutronix	2%	Google	1%
consultants	2%	Astaro	1%

Challenge

Out-of-tree code

Challenge

Out-of-tree code
Binary-only modules
Vendor-private code
External projects

Responses

Developer outreach

Merging outside projects
Even if the code isn't great
linux-staging tree

Discouraging binary modules

Challenge

Security

Challenge

Security
...of the kernel itself

Challenge

Security
...of the kernel itself
...support for user-space security

2008 CVEs (Jan - November)

CVE-2008-5033 CVE-2008-5029 CVE-2008-4934 CVE-2008-4933
CVE-2008-4618 CVE-2008-4576 CVE-2008-4554 CVE-2008-4445
CVE-2008-4410 CVE-2008-4395 CVE-2008-4302 CVE-2008-4210
CVE-2008-4113 CVE-2008-3915 CVE-2008-3911 CVE-2008-3901
CVE-2008-3889 CVE-2008-3833 CVE-2008-3832 CVE-2008-3831
CVE-2008-3792 CVE-2008-3686 CVE-2008-3535 CVE-2008-3534
CVE-2008-3528 CVE-2008-3527 CVE-2008-3526 CVE-2008-3525
CVE-2008-3496 CVE-2008-3276 CVE-2008-3275 CVE-2008-3272
CVE-2008-3247 CVE-2008-3077 CVE-2008-2931 CVE-2008-2826
CVE-2008-2812 CVE-2008-2750 CVE-2008-2729 CVE-2008-2372
CVE-2008-2365 CVE-2008-2358 CVE-2008-2148 CVE-2008-2137
CVE-2008-2136 CVE-2008-1675 CVE-2008-1673 CVE-2008-1669
CVE-2008-1619 CVE-2008-1615 CVE-2008-1375 CVE-2008-1367
CVE-2008-1294 CVE-2008-0600 CVE-2008-0598 CVE-2008-0352
CVE-2008-0010 CVE-2008-0009 CVE-2008-0007 CVE-2008-0001

Responses…?

User-space security

Unix-style DAC may not be enough

User-space security

In the mainline:
SELinux
SMACK

User-space security

Coming soon – maybe

AppArmor

TOMOYO Linux

TALPA / fanotify

Integrity management

Challenge

Scalability

Scalability issues

Locking

Contention kills performance

Cache effects hurt

Solutions

Finer-grained locking

Lockless algorithms

Scalability issues

Memory use

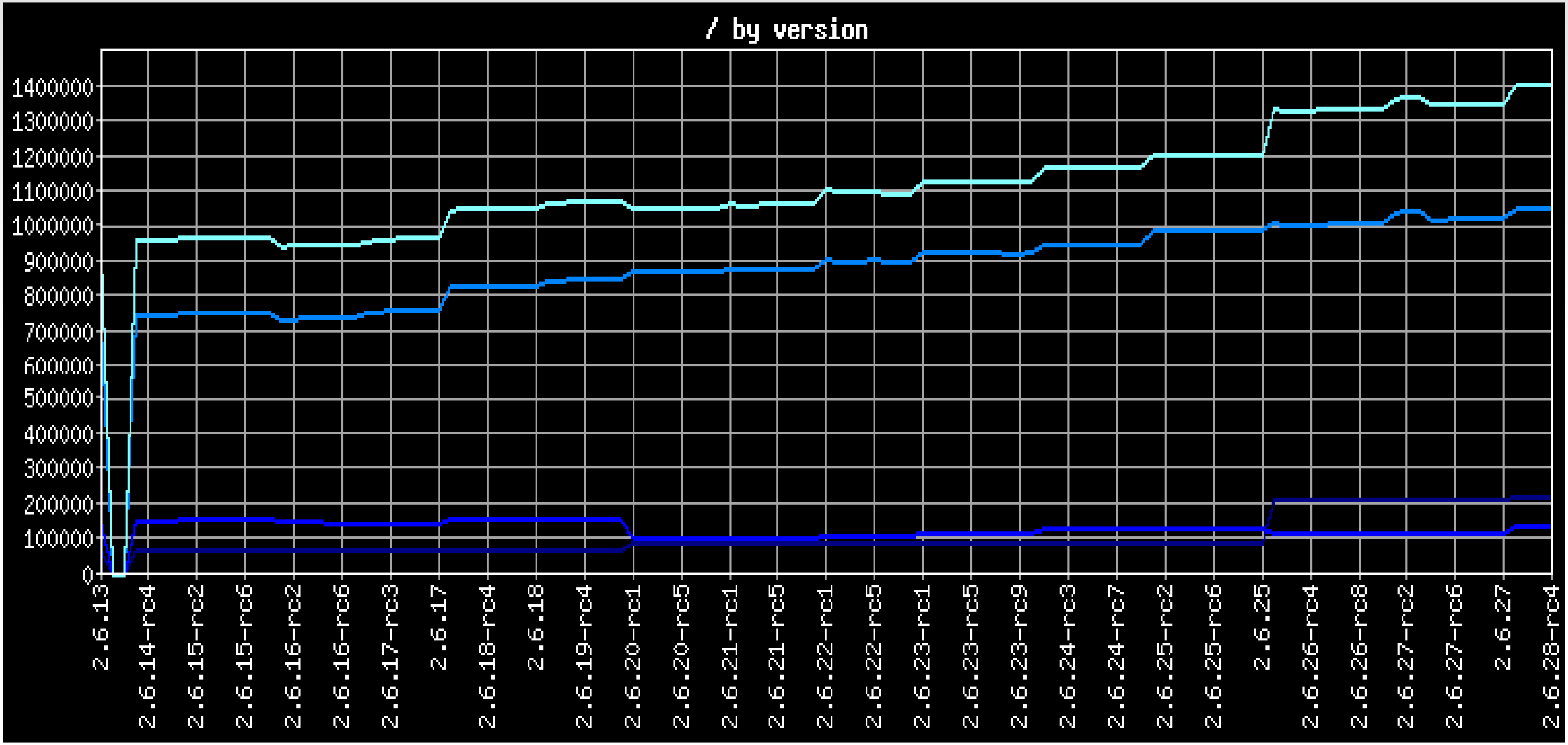
Scalability issues

Memory use

Solution: better data structures

Scalability goes both ways

Scalability issues



Scalability issues

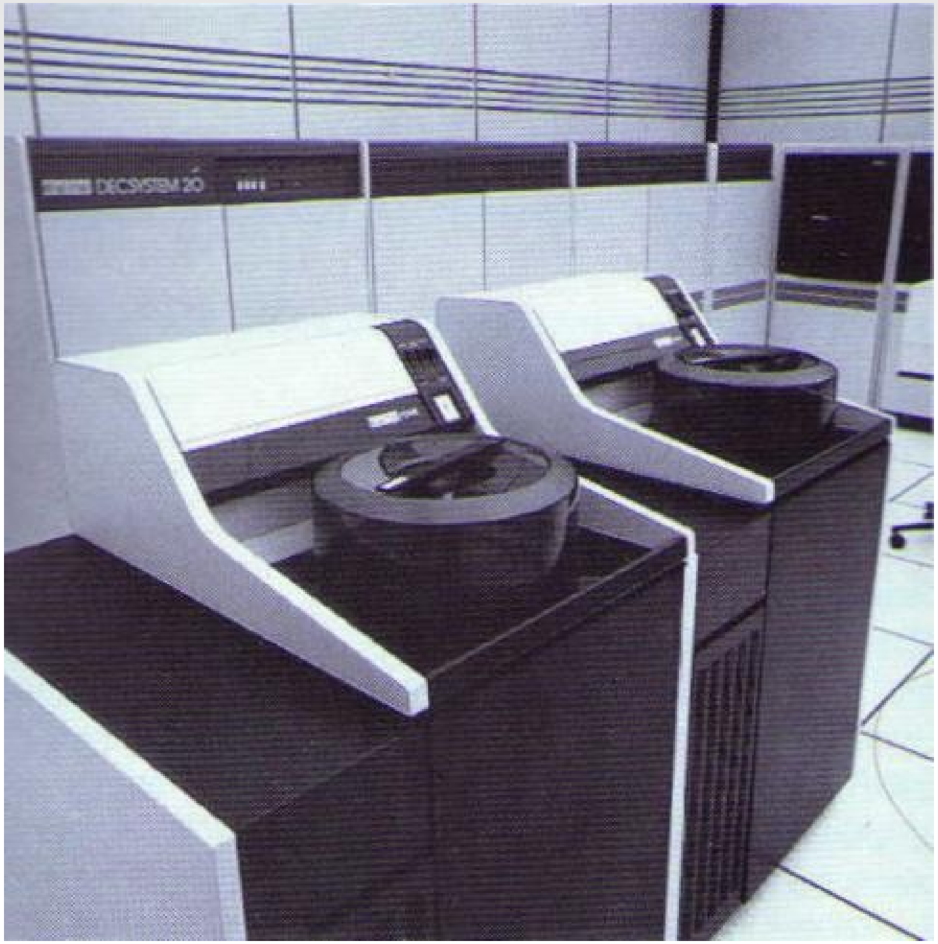
What to do?

More attention to bloat

More participation from embedded folks

Challenge

Storage and filesystems



Disks were small
…as were files

(DEC RP06, 178 MB)

Filesystem challenges

Inefficient metadata
fsck takes forever
Limits on file and filesystem sizes
No data integrity protection
Missing features

Generally old

Response: ext4

The progression of ext3
Extents
Better allocation
File and filesystem limits lifted
Journal checksums

Response: btrfs

A completely new filesystem

Extents

Subvolumes

Snapshots

Full checksumming

Fast fsck

Challenge

Solid-state storage
Truly random-access
Fast reads, slow writes
Wear leveling required

Our current flash filesystems
...are showing their age

Responses

Btrfs

UBIFS
Merged for 2.6.27
Expects direct access to flash

Logfs
Stalled for now

Challenge

Hardware support

Responses

Life just gets better

AMD/ATI releases information

Atheros hires community developers

VIA employs a community liaison

Sometimes life improves slowly

Wireless networking

Video adapters

Help life get better yet
Avoid closed hardware
Avoid binary-only drivers
Avoid uncooperative companies

Challenge

Hard real-time support

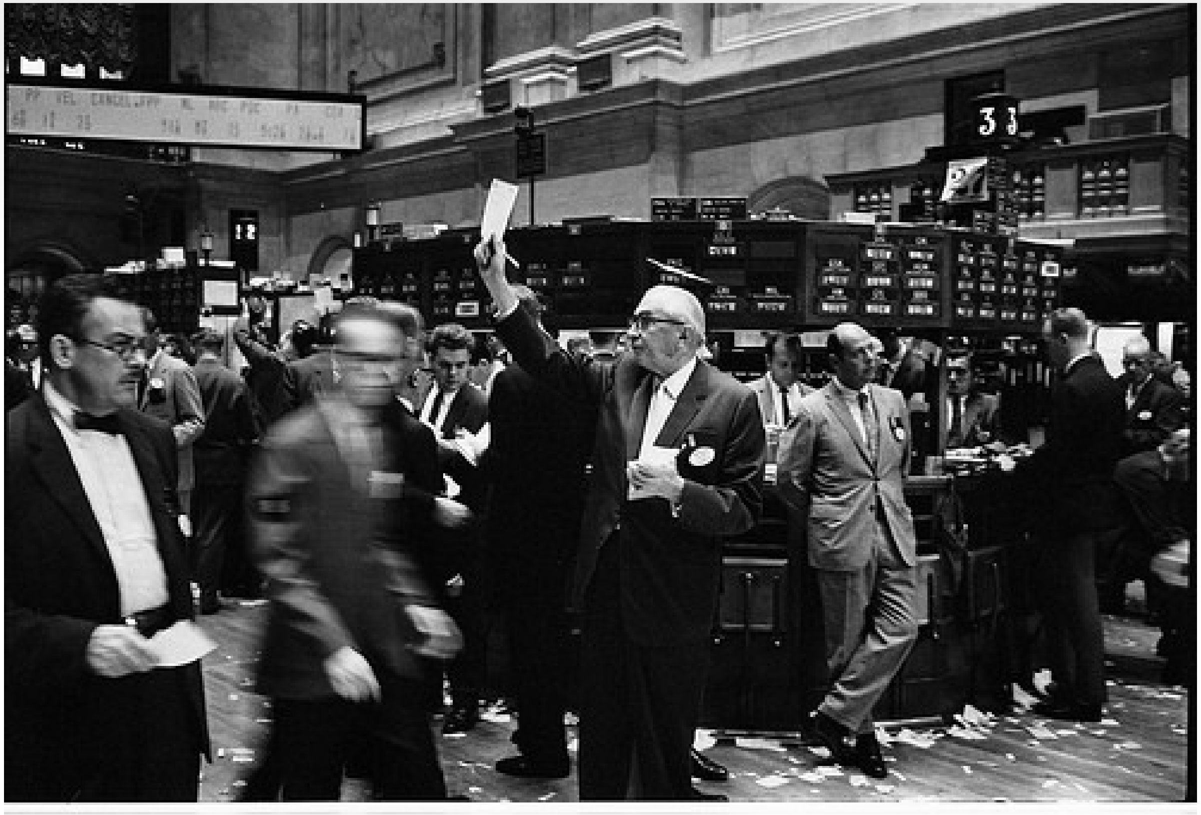
Who needs realtime?

Data acquisition / process control

DeLaval

Who needs realtime?

Commercial exchanges



Who needs realtime?

Gadgets

Realtime responses

The realtime patch set
Sleeping spinlocks
Threaded interrupt handlers
Lots of other stuff

Challenge

Containers

Responses

Much code already merged
Control groups
Resource controllers
Network, PID, user, ... namespaces

Some still waiting
Sysfs support
Checkpoint and restore
Management support

Challenge

Tracing

Responses

SystemTap
Powerful tool
Dynamic tracing
Painful to use
No user-space tracing

Responses

Ftrace
Simple, static tracing
2.6.27, lots of work in 2.6.28

LTTng
Complex static tracing

Trace buffer code
Common infrastructure for tracing
2.6.28

Why not just port DTrace?

Questions?