Seven challenges for the kernel community

(aka
The Kernel Report
LinuxTag 2010 edition)

Jonathan Corbet
LWN.net
corbet@lwn.net
1 Vitality
2 Scalability
3 Storage
4 Visibility
5 Response
6 Containment
7 Hardware
1: Vitality

Keeping the process strong and active
“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)
2.6.30 to 2.6.35-rc1+

(June 9, 2009 to June 4, 2010)

50,930 Changesets merged
  2,700 Developers contributed
  324 Employers represented

The kernel grew by 2.1 million lines
141 changesets per day

5,800 lines of code added every day (every day!)
<table>
<thead>
<tr>
<th>Source</th>
<th>Contribution</th>
<th>Source</th>
<th>Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>16.2%</td>
<td>Fujitsu</td>
<td>1.7%</td>
</tr>
<tr>
<td>Red Hat</td>
<td>11.9%</td>
<td>AMD</td>
<td>1.6%</td>
</tr>
<tr>
<td>unknown</td>
<td>7.9%</td>
<td>Texas Inst.</td>
<td>1.5%</td>
</tr>
<tr>
<td>Intel</td>
<td>7.7%</td>
<td>Atheros</td>
<td>1.4%</td>
</tr>
<tr>
<td>Novell</td>
<td>5.0%</td>
<td>academics</td>
<td>1.4%</td>
</tr>
<tr>
<td>IBM</td>
<td>4.8%</td>
<td>Analog Devices</td>
<td>1.4%</td>
</tr>
<tr>
<td>consultants</td>
<td>2.4%</td>
<td>HP</td>
<td>1.0%</td>
</tr>
<tr>
<td>Nokia</td>
<td>2.3%</td>
<td>Pengutronix</td>
<td>1.0%</td>
</tr>
<tr>
<td>Renesas Tech</td>
<td>2.0%</td>
<td>Wolfson Micro</td>
<td>1.0%</td>
</tr>
<tr>
<td>Oracle</td>
<td>1.9%</td>
<td>Broadcom</td>
<td>0.8%</td>
</tr>
</tbody>
</table>
2.6.30 (June 9)

TOMOYO Linux

Integrity measurement

R6xx/R7xx graphics support

Nilfs
2.6.31 (September 9)

Performance counter support

Char devices in user space

Kmemleak

TTM and Radeon KMS support

Storage topology
2.6.32 (December 2)

Lots of block scalability work

Performance-counter Perf events improvements

Scheduler interactivity work

Kernel Shared Memory

HWPOISON
2.6.33 (February 24)

Dynamic ftrace

DRBD distributed storage device

I/O bandwidth controller

TCP Cookie Transactions

Nouveau driver
2.6.34 (May 15)

Asynchronous suspend / resume

Better dynamic tracing (and tracing in general)

Better graphics support

LogFS flash filesystem

Ceph distributed filesystem
2.6.35 (August)

RPS/RFS (network scalability)

Memory compaction

“perf kvm” for combined host/guest monitoring

Lots of internal infrastructure work
Consensus:

It is working fairly well.
Participation

The kernel is a community-developed resource

It only works if we all cooperate!
2: Scalability
The scalability problem

Systems just get bigger
More CPUs
More memory
More I/O bandwidth
...

Problem areas

dcache_lock

Networking
  Can drive 10G at wire speed
  ...with large packets!

Solid state storage devices
  100,000 operations/second on the horizon
Scaling down

Scalability must work both ways
3: Storage
What's happening

Storage devices are getting larger
Not always faster

Usage patterns are changing

Solid-state storage devices are coming
ext4

Advantages
  Better performance
  Many limits lifted
  ext3 compatibility

Going into production use
Btrfs

A totally new filesystem

Advantages
Performance
Full checksumming
Snapshots
Internal volume management / RAID

Merged for 2.6.29
Still experimental
Btrfs is the default MeeGo filesystem
Solid-state storage

Rotating storage is dying
...well, maybe...

Solid-state devices are cool
Fast
Power-efficient
Shock-resistant
Solid-state storage

Also presents some challenges...

Poor performance
especially over time

Badly specified/implemented commands
TRIM
4: Visibility

Photo: Armel Genon
We want to know

What is the system doing?

Why are things slow?

Where should optimization effort be focused?

How do specific changes affect performance?
We want to know

What is the system doing?

Why are things slow?

Where should optimization effort be focused?

How do specific changes affect performance?

How can we shut up the DTrace fans?
SystemTap

A powerful dynamic tracing environment

Some problems
  Complex, difficult to use
  Requires lots of ancillary data
  Disconnect with kernel community
Ftrace

Lightweight kernel tracing facility
Popular with kernel developers

Lots of static tracing options
Dynamic tracing in 2.6.33

Where a lot of the action is
Perfcounter Perf Events

Access to performance monitoring registers
Useful for low-level optimization

Integrated with tracepoints

Lots happening in this area

Expect a merger with Ftrace eventually
“The kernel's whole approach to messaging is pretty haphazard and lame and sad. There have been various proposals to improve the usefulness and to rationally categorise things in way which are more useful to operators, but nothing seems to ever get over the line.”

-- Andrew Morton
LTTng

Linux Trace Toolkit
  Well-developed static tracing toolkit
  Extensive user-space tools
5: Response (realtime)

Realtime is about deterministic response

...not about speed
Sometimes the opposite of “real fast”
Real time

Used in gadgets
Realtime

Used in financial services

Photo: MojoBaer
The realtime preemption patch set

Deterministic realtime for Linux

Large, out-of-tree patch
   Shipped by numerous distributors
Realtime stuff in mainline

Threaded interrupt handlers

 Mutexes

 Priority inheritance

 Lots of latency reduction patches
What's left

Sleeping spinlocks
  Precursor work merged for 2.6.33

Problem areas
  Atomic kmaps
  Per-CPU variables
  Slab allocator

Big kernel lock removal
  Almost done in 2.6.35!
Deadline scheduling

Realtime people like to talk about deadlines
...rather than priorities

Thus: deadline scheduling
Three params:
   How much work to do
   When it must be done
   How often

SCHED_DEADLINE in the works
Lots details to deal with yet
6: Containment

Photo: photohome_uk
Two approaches

Virtualization
  Give each guest its own virtual machine
  Can run different operating systems

Containers
  Isolate guests running on the host kernel
  More efficient
  Trickier to implement
Virtualization

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

Remaining work: performance, management
Virtualized memory management

KSM (Kernel Samepage Merging)
  Scan for pages with duplicate contents
  Make everybody share one copy
  2.6.32

Compcache
  Swap out memory - to memory
  Compress on the way
  2.6.33 (staging)

Transcendent memory
  Waiting on the wings
Containers

In progress: namespace isolation
A multi-year project

In progress: resource controllers

 Longer-term: checkpoint/restart
Save state of container; restore later, maybe elsewhere
A difficult problem
7: Hardware

Photo: superde1uxe
Hardware support

Near universal

A few remaining problems
  Graphics
  Some network adapters

Some new problems
  Mobile devices
Power management

A problem for small systems
...and big systems too

Device-level support mostly done

What's left: system-level power management
Questions?
Have fun!