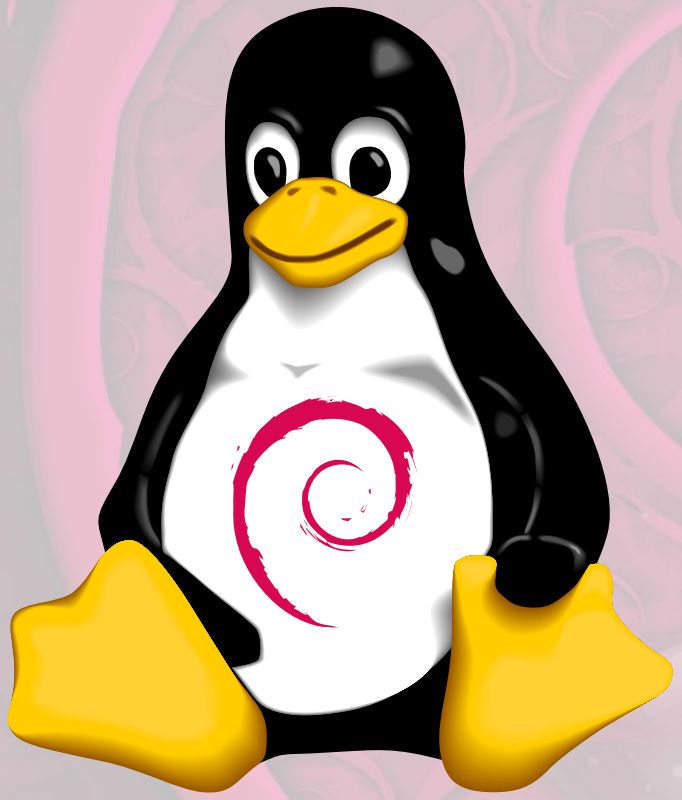
What's new in the Linux kernel

and what's missing in Debian



Ben Hutchings

mini-DebConf Cambridge 2017

Ben Hutchings

- Working on kernel stuff for Debian and in paid work for about 10 years
- Debian kernel and LTS team member, now doing most of the kernel maintenance aside from ports
- Maintaining Linux 3.2.y and 3.16.y stable update series on kernel.org

Linux releases early and often

- Linux is released about 5 times a year (plus stable updates every week or two)
  - ...though some features aren't ready to use when they first appear in a release
- Since my talk at DebConf last year, Linus has made 8 releases (4.7-4.14)
- Good news: we have lots of new kernel features in testing/unstable
- Bad news: some of them won't really work without new userland

Recap of last year's features

- User-space page fault handling
  - Used by QEMU in stretch for post-copy migration
  - Also used by CRIU, not included in stretch
- Lightweight tunnels
  - Supported by iproute2 in stretch
- Raspberry Pi
  - Pi 2 (armhf) and 3 (arm64) supported in stretch
  - Still hard to install, and stretch kernel lacks wireless drivers
- Kernel hardening
  - Several hardening features enabled for stretch
- Real-Time Linux
  - Included in stretch, but still requires a large patch set

Tracing enhancements [4.7-4.9]

- Static tracepoints provide detailed, structured information about what kernel is doing
- Stream of trace events had to be copied and analysed in user-space – can have significant performance cost
- Now possible to do some analysis in kernel – similar to SystemTap or DTrace
- Tracepoint programs run in BPF VM – constrains use of CPU and memory
- Tracepoint programs can update map structures visible to user space
- Also usable with kprobes and uprobes
- bpfcc compiles subset-of-C to BPF and comes with many useful tracepoint programs

Kernel hardening [ongoing]

- KSPP porting and reimplementing some features from PaX/Grsecurity
- kASLR implemented on MIPS and extended on x86
- SLAB and SLUB support free-list randomisation – makes use-after-free harder to exploit
- Hardened user-copy adds bounds check on copies between user and kernel space
- Virtually mapped stack on amd64
- All of the above enabled in stretch
- Enabled IO_STRICT_DEVMEM – stops user space interfering with kernel drivers
- Enabled SECURITY_DMESG_RESTRICT – stops unprivileged users reading sensitive kernel data

Reorganised docs [ongoing]

- DocBook replaced by reStructuredText (RST) processed by Sphinx
- Many plain text docs converted to RST
- Many docs moved into subdirectories for users/admins, user-space developers, kernel developers, etc.
- All RST docs converted to hierarchical, searchable HTML pages
- Demo: searching linux-doc
- No more manual pages for kernel API

schedutil [4.7]

- Scheduler knows (roughly) speed of CPUs and uses it to decide where task should run
- CPUFreq governors like ondemand monitor how busy CPU is, to decide target speed
- Scheduler doesn't know when governor has reduced CPU speed, and governor takes time to work out that higher speed is needed
- This mostly still works, but governor doesn't respond as quickly as it should
- schedutil is a new CPUFreq governor that uses more information from the scheduler to decide the target speed
- Not yet used by default
  - Use cpupower to select it
  - For Intel CPUs, add kernel param `intel_pstate=passive`

Zoned recording [4.7-4.13] (1)

- Hard drive capacity constrained by size of write head – can't shrink it further
- Shingled magnetic recording (SMR) increases capacity by grouping tracks into “zones” where writes partly overlap

Zoned recording [4.7-4.13] (2)

- Each zone must be rewritten in order, and tracks after the last written may be unreadable
- This requires a translation layer, similar to flash – either drive-managed (compatible, but maybe slow) or host-managed
- Zones supported on ATA [4.7] and SCSI [4.10] drives
- dm-zoned [4.13] implements translation layer for host-managed drives
  - RFP for dm-zoned-tools: #882640

Block layer replacement [ongoing]

- Block layer handles request queues for storage devices
- Old (2.6) block layer has performance limitations – like single queue – and other problems
- blk-mq introduced [3.13] to solve these problems – and worked well for fast devices
- It didn't work so well for slower devices – they need a scheduler to manage and prioritise the queue
- Three I/O schedulers now available – deadline [4.11], bfq [4.12], kyber [4.12]
- SCSI subsystem (also used for ATA and USB) can use it but requires module param to opt-in
- Several block drivers not yet converted – most important is MMC, likely to be done in 4.16

eXpress Data Path [4.8-4.14]

- XDP is optional step in network receive path for early filtering – yet another application of BPF
- May be implemented by driver or even hardware – the earlier the better
- Filter program may drop packet, pass it up or modify and send it back through same interface
- Useful for:
  - DoS mitigation
  - Load balancing
  - Network monitoring
- Can replace some uses of user-level networking

Graphics on ARM [ongoing]

- New drivers for Allwinner, Amlogic, ARM, HiSilicon, Mediatek, STM, ZTE display controllers
- Continuing work on drivers for Broadcom (vc4), Nvidia (tegra), Qualcomm (msm) and Vivante (etnaviv) GPUs
  - All now supported by Mesa in unstable
- No progress on ARM Mali GPU used by many SoC vendors

statx() [4.11]

- stat(), fstat(), lstat() system calls fill in a `struct stat` with metadata about a file
- Metadata extended repeatedly, requiring new structure and syscall – see stat(2)
- Getting all the metadata can be expensive, but most callers don't need all of it – e.g. `ls --color` only needs mode bits
- Other callers wanted extra metadata (creation time, attribute flags, ...)
- New statx() syscall fills in a `struct statx`, but also takes flags specifying which fields are really wanted
- Additional flags to control use of cached metadata for network filesystems
- Y2038 safe on 32-bit architectures
- Not yet supported by glibc

Firmware removal [4.14]

- All the blobs in `firmware/` have been removed upstream
- ...but still a few non-free bits elsewhere in the source tree (stripped from Debian package)

Packaging changes

- Added linux-signed package and modified linux to support signed kernels and modules
  - ...but the signing infrastructure isn't ready, so this was reverted
- Fixed cross-building (except for tools/perf)
- Added debug symbols on all architectures
- Changed linux-headers-*abiname*-common to an arch-independent package – needed for arm64 and useful for cross-building
- Removed xen-linux-system-*abiname-flavour* and linux-manual-*version* packages

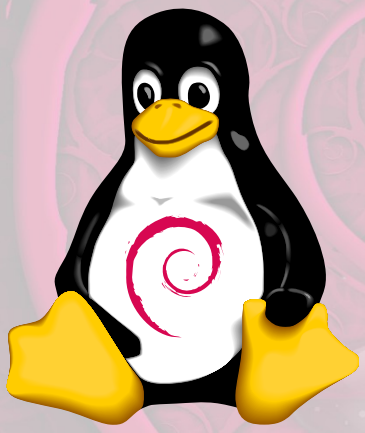
Questions?

Credits

- Linux 'Tux' logo © Larry Ewing, Simon Budig.
  - Modified by Ben to add Debian open-ND logo
- Debian open-ND logo © Software in the Public Interest, Inc.
- Debian slide template © Raphaël Hertzog
- Background image © Alexis Younes

What's new in the Linux kernel

and what's missing in Debian



Ben Hutchings

mini-DebConf Cambridge 2017

Ben Hutchings

- Working on kernel stuff for Debian and in paid work for about 10 years
- Debian kernel and LTS team member, now doing most of the kernel maintenance aside from ports
- Maintaining Linux 3.2.y and 3.16.y stable update series on kernel.org

Linux releases early and often

- Linux is released about 5 times a year (plus stable updates every week or two)
  - ...though some features aren't ready to use when they first appear in a release
- Since my talk at DebConf last year, Linus has made 8 releases (4.7-4.14)
- Good news: we have lots of new kernel features in testing/unstable
- Bad news: some of them won't really work without new userland

Recap of last year's features

- User-space page fault handling
  - Used by QEMU in stretch for post-copy migration
  - Also used by CRIU, not included in stretch
- Lightweight tunnels
  - Supported by iproute2 in stretch
- Raspberry Pi
  - Pi 2 (armhf) and 3 (arm64) supported in stretch
  - Still hard to install, and stretch kernel lacks wireless drivers
- Kernel hardening
  - Several hardening features enabled for stretch
- Real-Time Linux
  - Included in stretch, but still requires a large patch set

Tracing enhancements [4.7-4.9]

- Static tracepoints provide detailed, structured information about what kernel is doing
- Stream of trace events had to be copied and analysed in user-space – can have significant performance cost
- Now possible to do some analysis in kernel – similar to SystemTap or DTrace
- Tracepoint programs run in BPF VM – constrains use of CPU and memory
- Tracepoint programs can update map structures visible to user space
- Also usable with kprobes and uprobes
- bpfcc compiles subset-of-C to BPF and comes with many useful tracepoint programs

Kernel hardening [ongoing]

- KSPP porting and reimplementing some features from PaX/Grsecurity
- kASLR implemented on MIPS and extended on x86
- SLAB and SLUB support free-list randomisation – makes use-after-free harder to exploit
- Hardened user-copy adds bounds check on copies between user and kernel space
- Virtually mapped stack on amd64
- All of the above enabled in stretch
- Enabled IO_STRICT_DEVMEM – stops user space interfering with kernel drivers
- Enabled SECURITY_DMESG_RESTRICT – stops unprivileged users reading sensitive kernel data

Reorganised docs [ongoing]

- DocBook replaced by reStructuredText (RST) processed by Sphinx
- Many plain text docs converted to RST
- Many docs moved into subdirectories for users/admins, user-space developers, kernel developers, etc.
- All RST docs converted to hierarchical, searchable HTML pages
- Demo: searching linux-doc
- No more manual pages for kernel API

schedutil [4.7]

- Scheduler knows (roughly) speed of CPUs and uses it to decide where task should run
- CPUFreq governors like ondemand monitor how busy CPU is, to decide target speed
- Scheduler doesn't know when governor has reduced CPU speed, and governor takes time to work out that higher speed is needed
- This mostly still works, but governor doesn't respond as quickly as it should
- schedutil is a new CPUFreq governor that uses more information from the scheduler to decide the target speed
- Not yet used by default
  - Use cpupower to select it
  - For Intel CPUs, add kernel param `intel_pstate=passive`

Zoned recording [4.7-4.13] (1)

- Hard drive capacity constrained by size of write head – can't shrink it further
- Shingled magnetic recording (SMR) increases capacity by grouping tracks into “zones” where writes partly overlap

Zoned recording [4.7-4.13] (2)

- Each zone must be rewritten in order, and tracks after the last written may be unreadable
- This requires a translation layer, similar to flash – either drive-managed (compatible, but maybe slow) or host-managed
- Zones supported on ATA [4.7] and SCSI [4.10] drives
- dm-zoned [4.13] implements translation layer for host-managed drives
  - RFP for dm-zoned-tools: #882640

Block layer replacement [ongoing]

- Block layer handles request queues for storage devices
- Old (2.6) block layer has performance limitations – like single queue – and other problems
- blk-mq introduced [3.13] to solve these problems – and worked well for fast devices
- It didn't work so well for slower devices – they need a scheduler to manage and prioritise the queue
- Three I/O schedulers now available – deadline [4.11], bfq [4.12], kyber [4.12]
- SCSI subsystem (also used for ATA and USB) can use it but requires module param to opt-in
- Several block drivers not yet converted – most important is MMC, likely to be done in 4.16

eXpress Data Path [4.8-4.14]

- XDP is optional step in network receive path for early filtering – yet another application of BPF
- May be implemented by driver or even hardware – the earlier the better
- Filter program may drop packet, pass it up or modify and send it back through same interface
- Useful for:
  - DoS mitigation
  - Load balancing
  - Network monitoring
- Can replace some uses of user-level networking

Graphics on ARM [ongoing]

- New drivers for Allwinner, Amlogic, ARM, HiSilicon, Mediatek, STM, ZTE display controllers
- Continuing work on drivers for Broadcom (vc4), Nvidia (tegra), Qualcomm (msm) and Vivante (etnaviv) GPUs
  - All now supported by Mesa in unstable
- No progress on ARM Mali GPU used by many SoC vendors

statx() [4.11]

- stat(), fstat(), lstat() system calls fill in a `struct stat` with metadata about a file
- Metadata extended repeatedly, requiring new structure and syscall – see stat(2)
- Getting all the metadata can be expensive, but most callers don't need all of it – e.g. `ls --color` only needs mode bits
- Other callers wanted extra metadata (creation time, attribute flags, ...)
- New statx() syscall fills in a `struct statx`, but also takes flags specifying which fields are really wanted
- Additional flags to control use of cached metadata for network filesystems
- Y2038 safe on 32-bit architectures
- Not yet supported by glibc

Firmware removal [4.14]

- All the blobs in `firmware/` have been removed upstream
- ...but still a few non-free bits elsewhere in the source tree (stripped from Debian package)

Packaging changes

- Added linux-signed package and modified linux to support signed kernels and modules
  - ...but the signing infrastructure isn't ready, so this was reverted
- Fixed cross-building (except for tools/perf)
- Added debug symbols on all architectures
- Changed linux-headers-*abiname*-common to an arch-independent package – needed for arm64 and useful for cross-building
- Removed xen-linux-system-*abiname-flavour* and linux-manual-*version* packages


Questions?
debian