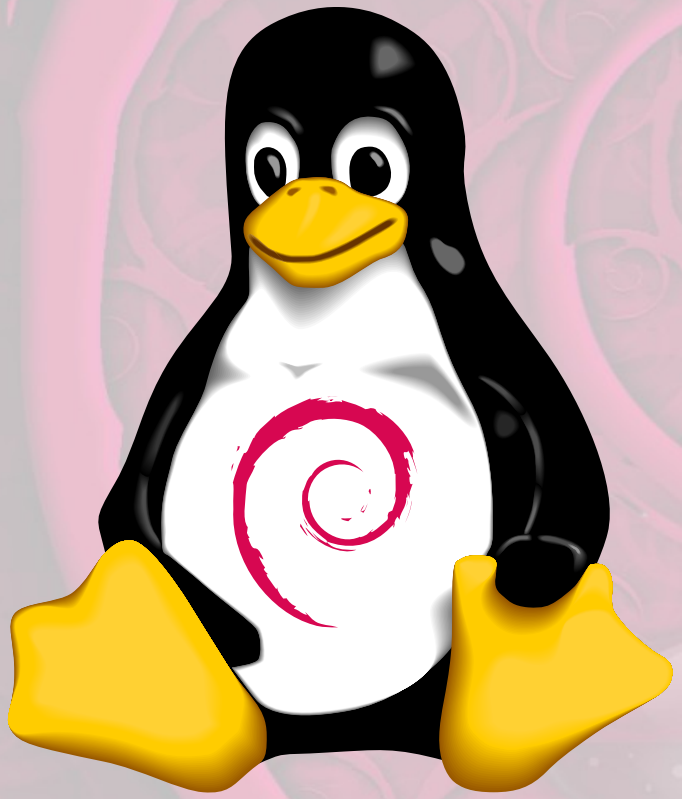
What's new in the Linux kernel

and what's missing in Debian



Ben Hutchings
DebConf 16

Ben Hutchings

- Professional software engineer by day, Debian developer by night (or sometimes the other way round)
- Regular Linux contributor in both roles since 2008
- Working on various drivers and kernel code in my day job
- 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 last year, Linus has made 5 releases (4.2-4.6)
- 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 (1)

- Extended Berkeley Packet Filter (eBPF):
  - further extensions, bpf() system call and filesystem added
  - verifier stops programs leaking information about kernel memory layout, so now all users can use eBPF
  - now supported as a target in LLVM
  - extensions still not yet widely used
  - JIT still not enabled by default; needs changes so it can't be used to create 'gadgets' for privilege escalation [ongoing]
- overlayfs:
  - now works on top of NFS, so can be used by FAI and LTSP
  - other limitations still exist
- atomic mode-setting: supported on some more ARM SoCs, but still not used by Xorg or Wayland

Recap of last year's features (2)

- live patching: some interest in this, but no progress in Debian yet
- non-volatile DIMMs:
  - DAX support added to XFS
  - new kernel infrastructure: libnvdimm
  - missing ndctl management utility (RFP: #829257)
- ext4 encryption: not supported in the installer – should it be?
- Intel MPX: ready to use?
- batched network transmit: supported in more drivers, no userland changes needed
- Y2038 compliance: some in-kernel APIs fixed; no userland ABI changes yet

New cgroup controllers [4.2,4.3]

- Writeback controller allows fairer sharing of I/O bandwidth for buffered writes
  - Buffering writes is essential, but buffering too much is a problem
  - Block I/O controller couldn't share out bandwidth because writeback I/O wasn't associated with a process
  - Memory controller couldn't throttle writers when necessary because it didn't know anything about I/O bandwidth
  - Writeback controller does a better job, by tracking which process is most responsible for writing to each file
  - Requires help from the specific filesystem – currently only implemented for btrfs, ext2, ext4
- PIDs controller allows limiting the number of processes
  - Each PID namespace has limited PIDs – can be 2^{31}-1 but is usually 32767 for compatibility
  - PIDs controller can prevent exhaustion of PIDs by accident or malice

User-space page fault handling [4.3]

- “Anonymous” memory (not file-backed) can be swapped out; access causes page fault and kernel swaps it in
- Live migration of VM or container moves its anonymous memory in one of two ways:
  - Pre-copy: start copying with VM/container still running on source; freeze it when remaining pages are changed too quickly to copy this way; finish copying; resume on destination – can be very slow
  - Post-copy: freeze VM/container on source; start copying; resume on destination; finish copying – can be more efficient but needs different page fault handling for unmigrated pages
- userfaultfd() and related ioctls allow user-space to override page fault handling for address ranges
- QEMU/KVM uses this to implement post-copy live migration
- CRIU will likely use it in future

Lightweight tunnels [4.3]

- Tunnel devices:
  1. Create device, configured to {en,de}capsulate packets transferred via existing device or address
  2. Create route via tunnel device
- Lightweight tunnel:
  1. Create route via existing device or address, configured to {en,de}capsulate packets
- Encapsulations supported: IPv4, IPv6, ILA, MPLS
- Needs iproute2 v4.4+, not yet in Debian (#829305)

ARM soft PAN [4.3]

- Kernel should only access user-space memory through specific safe functions
- Accidental access to user-space from another function is often exploitable for privilege escalation
- Some recent CPUs have feature to mitigate this (Intel: 'SMAP'; ARM: 'PAN') – turns an 'pwn' into an 'oops'
- ARMv7 doesn't include PAN... but does include 'domains' feature that can be used to do the same thing

Reproducible builds [4.3-4.4]

- Kernel and modules already reproducible, if $KBUILD_BUILD_TIMESTAMP set properly
- Documentation was not – included current date, randomised IDs, randomised hash ordering, ...
- Changes accepted upstream to fix all of these issues

Raspberry Pi [3.7-4.5]

- Series of low-cost development boards using Broadcom VideoCore SoCs
- VideoCore architecture is proprietary, but SoCs also include ARM core(s)
- Default OS for the ARM side is Debian derivative (Raspbian) with heavily patched kernel
- Drivers and platform code have gradually been cleaned up and merged upstream over past 4 years
  - GPU drivers rewritten to run on ARM instead of VPU
- Raspberry Pi 2 supported in Debian starting with linux 4.4~rc8-1~exp1 and flash-kernel 3.62

Kernel hardening [ongoing]

- Kernel Self-Protection Project is porting hardening features from PaX and Grsecurity ... gradually
- Less writeable data [4.6]:
  - Write-protection enforced by default on more architectures
  - Data can be write-protected after initialisation code runs
- Page poisoning [4.6]:
  - Free memory is still accessible, still contains old values, and may be reused soon
  - Use-after-free bugs often exploitable for information leak or privilege escalation
  - Page poisoning trashes free memory – already available as a debug feature; cheaper option available as mitigation
- GCC plugins [ongoing]

Real-Time Linux [ongoing]

- Real-Time Linux project adds compile-time option (PREEMPT_RT) that limits scheduling latency
  - This is about *worst-case* latency, not average latency – which typically gets worse
- Developed as long-lived fork, but many changes have been merged into mainline
- Briefly wound down due to lack of funding, but Linux Foundation now paying main developer (Thomas Gleixner)
- Patch series released for Linux 4.4.y and 4.6.y
- More changes going into mainline:
  - Timer wheel rework [4.2]
  - CPU hotplug rework [4.6-4.7]

Packaging changes

- Binary packages are reproducible
- linux package supports stage1 build profile for architecture bootstrapping
- linux and linux-tools packages combined, with build profile to exclude tools packages
- linux package can be configured to disable some binaries in derivative packages (like linux-grsec)
- Preparation for Secure Boot support – module signing, kernel image signing, securelevel
- Building lockdep and cpupower packages
- Installer includes drivers by directory, not just by name
- Dropped support for 586 and MIPS R1
- Rewrote maintainer scripts

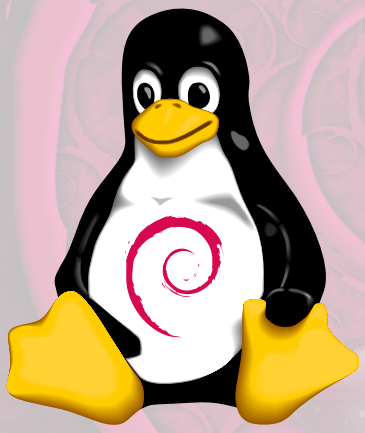
Questions?

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
DebConf 16

Ben Hutchings

- Professional software engineer by day, Debian developer by night (or sometimes the other way round)
- Regular Linux contributor in both roles since 2008
- Working on various drivers and kernel code in my day job
- 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 last year, Linus has made 5 releases (4.2-4.6)
- 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 (1)

- Extended Berkeley Packet Filter (eBPF):
  - further extensions, bpf() system call and filesystem added
  - verifier stops programs leaking information about kernel memory layout, so now all users can use eBPF
  - now supported as a target in LLVM
  - extensions still not yet widely used
  - JIT still not enabled by default; needs changes so it can't be used to create 'gadgets' for privilege escalation [ongoing]
- overlayfs:
  - now works on top of NFS, so can be used by FAI and LTSP
  - other limitations still exist
- atomic mode-setting: supported on some more ARM SoCs, but still not used by Xorg or Wayland

Recap of last year's features (2)

- live patching: some interest in this, but no progress in Debian yet
- non-volatile DIMMs:
  - DAX support added to XFS
  - new kernel infrastructure: libnvdimm
  - missing ndctl management utility (RFP: #829257)
- ext4 encryption: not supported in the installer – should it be?
- Intel MPX: ready to use?
- batched network transmit: supported in more drivers, no userland changes needed
- Y2038 compliance: some in-kernel APIs fixed; no userland ABI changes yet

New cgroup controllers [4.2,4.3]

- Writeback controller allows fairer sharing of I/O bandwidth for buffered writes
  - Buffering writes is essential, but buffering too much is a problem
  - Block I/O controller couldn't share out bandwidth because writeback I/O wasn't associated with a process
  - Memory controller couldn't throttle writers when necessary because it didn't know anything about I/O bandwidth
  - Writeback controller does a better job, by tracking which process is most responsible for writing to each file
  - Requires help from the specific filesystem – currently only implemented for btrfs, ext2, ext4
- PIDs controller allows limiting the number of processes
  - Each PID namespace has limited PIDs – can be 2^{31}-1 but is usually 32767 for compatibility
  - PIDs controller can prevent exhaustion of PIDs by accident or malice

User-space page fault handling [4.3]

- “Anonymous” memory (not file-backed) can be swapped out; access causes page fault and kernel swaps it in
- Live migration of VM or container moves its anonymous memory in one of two ways:
  - Pre-copy: start copying with VM/container still running on source; freeze it when remaining pages are changed too quickly to copy this way; finish copying; resume on destination – can be very slow
  - Post-copy: freeze VM/container on source; start copying; resume on destination; finish copying – can be more efficient but needs different page fault handling for unmigrated pages
- userfaultfd() and related ioctls allow user-space to override page fault handling for address ranges
- QEMU/KVM uses this to implement post-copy live migration
- CRIU will likely use it in future

Lightweight tunnels [4.3]

- Tunnel devices:
  1. Create device, configured to {en,de}capsulate packets transferred via existing device or address
  2. Create route via tunnel device
- Lightweight tunnel:
  1. Create route via existing device or address, configured to {en,de}capsulate packets
- Encapsulations supported: IPv4, IPv6, ILA, MPLS
- Needs iproute2 v4.4+, not yet in Debian (#829305)

ARM soft PAN [4.3]

- Kernel should only access user-space memory through specific safe functions
- Accidental access to user-space from another function is often exploitable for privilege escalation
- Some recent CPUs have feature to mitigate this (Intel: 'SMAP'; ARM: 'PAN') – turns an 'pwn' into an 'oops'
- ARMv7 doesn't include PAN... but does include 'domains' feature that can be used to do the same thing

Reproducible builds [4.3-4.4]

- Kernel and modules already reproducible, if $KBUILD_BUILD_TIMESTAMP set properly
- Documentation was not – included current date, randomised IDs, randomised hash ordering, ...
- Changes accepted upstream to fix all of these issues

Raspberry Pi [3.7-4.5]

- Series of low-cost development boards using Broadcom VideoCore SoCs
- VideoCore architecture is proprietary, but SoCs also include ARM core(s)
- Default OS for the ARM side is Debian derivative (Raspbian) with heavily patched kernel
- Drivers and platform code have gradually been cleaned up and merged upstream over past 4 years
  - GPU drivers rewritten to run on ARM instead of VPU
- Raspberry Pi 2 supported in Debian starting with linux 4.4~rc8-1~exp1 and flash-kernel 3.62

Kernel hardening [ongoing]

- Kernel Self-Protection Project is porting hardening features from PaX and Grsecurity ... gradually
- Less writeable data [4.6]:
  - Write-protection enforced by default on more architectures
  - Data can be write-protected after initialisation code runs
- Page poisoning [4.6]:
  - Free memory is still accessible, still contains old values, and may be reused soon
  - Use-after-free bugs often exploitable for information leak or privilege escalation
  - Page poisoning trashes free memory – already available as a debug feature; cheaper option available as mitigation
- GCC plugins [ongoing]

Real-Time Linux [ongoing]

- Real-Time Linux project adds compile-time option (PREEMPT_RT) that limits scheduling latency
  - This is about *worst-case* latency, not average latency – which typically gets worse
- Developed as long-lived fork, but many changes have been merged into mainline
- Briefly wound down due to lack of funding, but Linux Foundation now paying main developer (Thomas Gleixner)
- Patch series released for Linux 4.4.y and 4.6.y
- More changes going into mainline:
  - Timer wheel rework [4.2]
  - CPU hotplug rework [4.6-4.7]

Packaging changes

- Binary packages are reproducible
- linux package supports stage1 build profile for architecture bootstrapping
- linux and linux-tools packages combined, with build profile to exclude tools packages
- linux package can be configured to disable some binaries in derivative packages (like linux-grsec)
- Preparation for Secure Boot support – module signing, kernel image signing, securelevel
- Building lockdep and cpupower packages
- Installer includes drivers by directory, not just by name
- Dropped support for 586 and MIPS R1
- Rewrote maintainer scripts


Questions?
debian