



# *CE Workgroup*

## **Status of Embedded Linux**

**June 2022**

Tim Bird

Principal Software Engineer, Sony Electronics



CE Workgroup

# Nature of this talk...

- I periodically take a look at the status of embedded Linux
- This talk will be a weird hybrid of:
  - Talk on kernel technologies and industry news I give quarterly at Japan Technical Jamborees
  - Thoughts on the overall status of major technology areas and the embedded Linux ecosystem
- Hope to accomplish 2 things:
  - Let people know what's going on
  - Open a discussion on 'what's next' (and what needs more work)



CE Workgroup

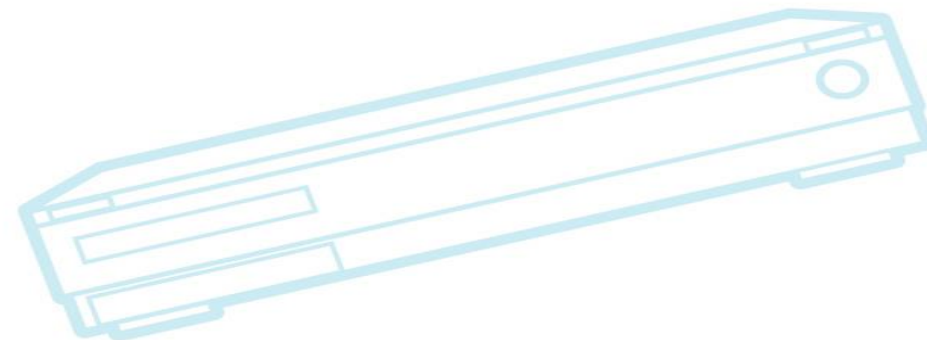
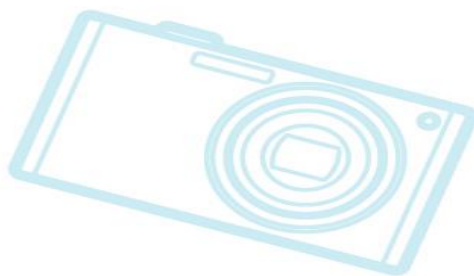
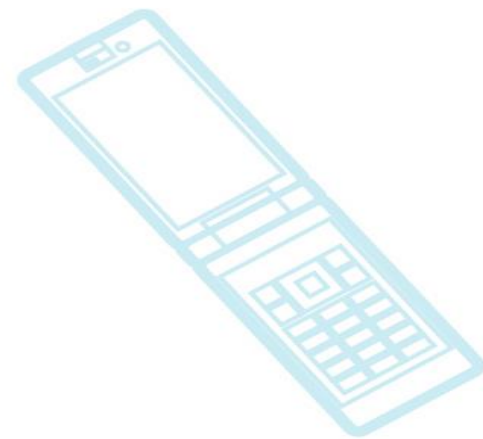
# Embedded Linux past vs. present



CE Workgroup

# Overall agenda

- History (past)
- Status (present)
- Are we done yet?







CE Workgroup

# A long time ago...

- In 1998, I started working on embedded Linux at Lineo
- In 2003, I was hired by Sony to help start the “Consumer Electronics Linux Forum” (CELF)
  - In 2005, CELF started the Embedded Linux Technical Conference
- In 2010, CELF became part of the Linux Foundation that became the “Core Embedded Linux Project” (CELP)



# In the early days...

- In the '90s we spent lots of effort convincing companies that Linux was suitable for embedded devices
  - Competition back then was: VRTX, VxWorks, pSOS, Nucleus, µltron (ie Commercial RTOSes)
- Architecture support: Intel, ARM, MIPS
  - But, no major chip vendor wrote arch support for Linux
  - It was all community developers, or nascent embedded Linux companies
- Embedded distros: Roll your own, or MontaVista, Lineo, TimeSys



# Focus areas back then

- CELF spent time analyzing and trying to address deficiencies in Linux, to make it suitable for embedded
- In 2003, CELF identified 5 key areas that needed work:
  - System size
  - Boot time
  - Power management
  - Realtime
  - Security
- CELF funded the contribution of several features in the Linux kernel in these and other areas (like filesystems)



CE Workgroup

# Are we done yet?

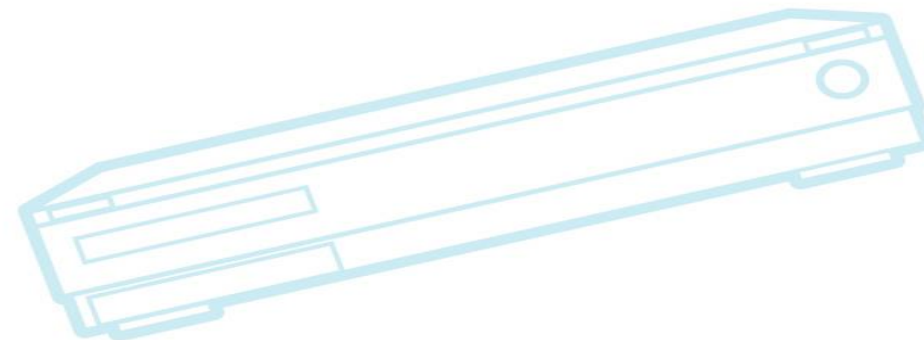
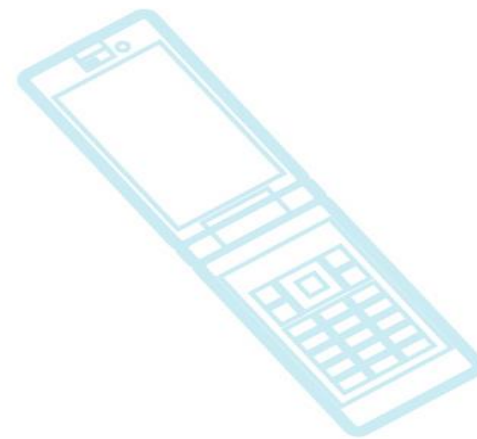
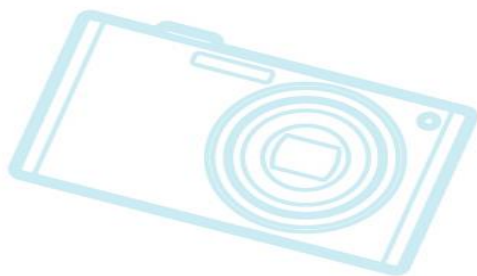
- I am still haunted by a phrase said in jest (I think) at ELC 2008 by Andrew Morton
  - “Are we done yet?”
- BTW: Andrews’ talk that year is available, and it still applicable for companies just getting into embedded Linux
  - See <https://bootlin.com/pub/video/2008/elc/elc2008-andrew-morton-keynote.ogg>





CE Workgroup

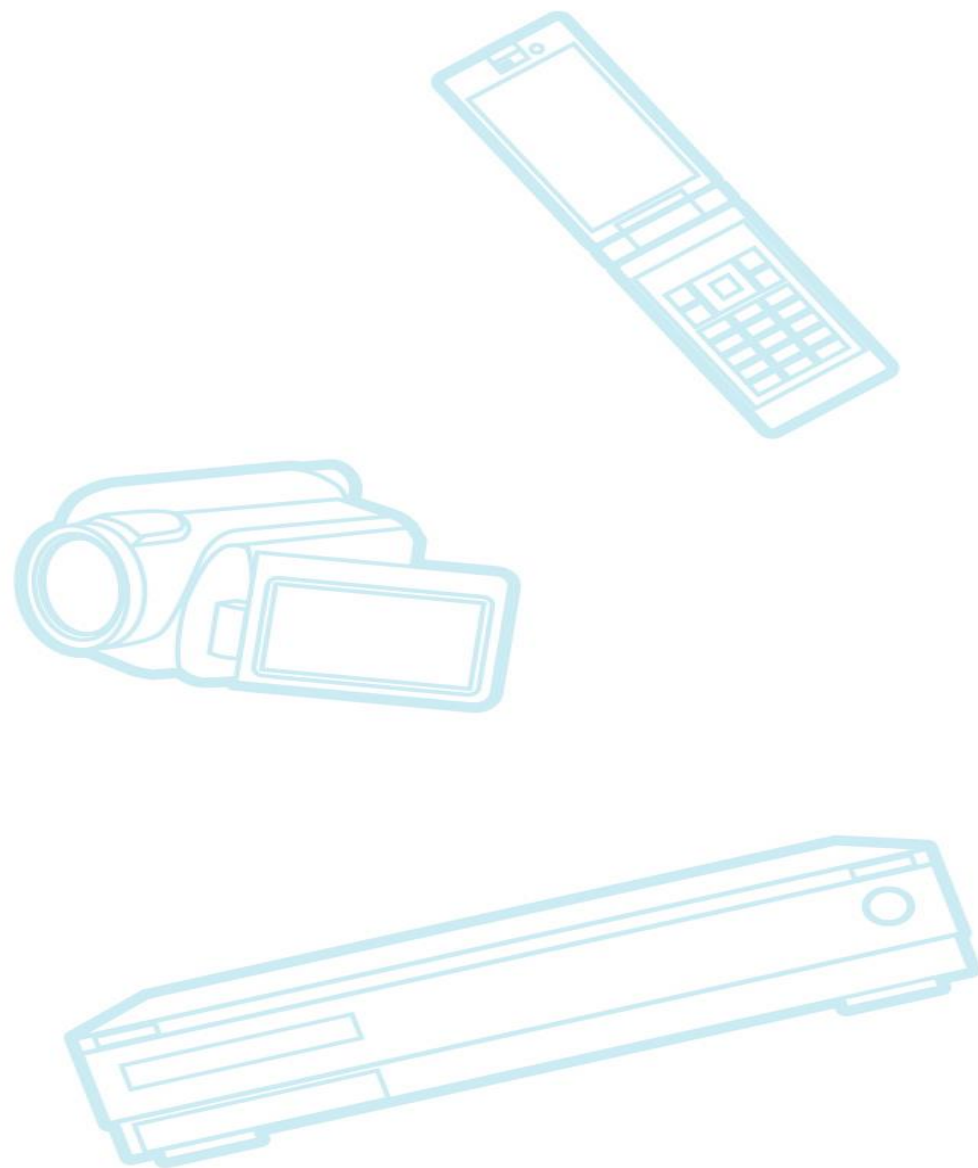
# Present: Linux Today





# Outline

Linux Kernel  
Technology Areas  
Industry News  
Scorecards  
Conclusions





CE Workgroup

# Outline

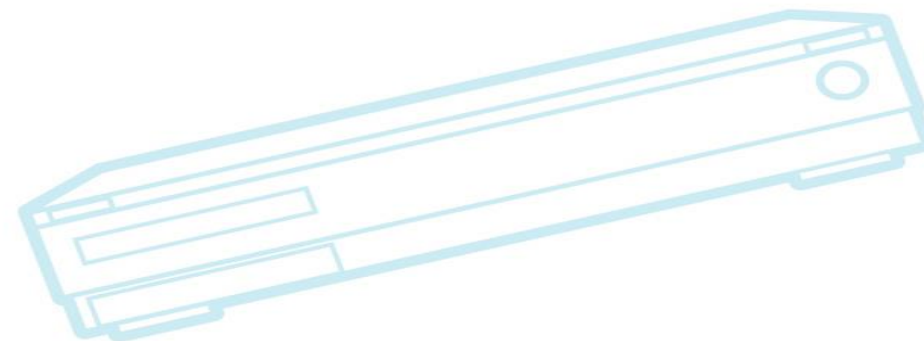
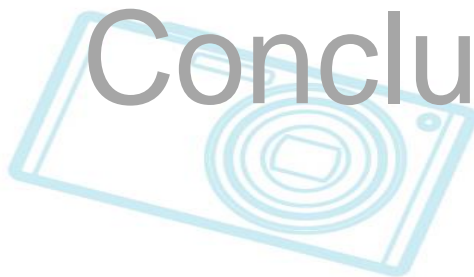
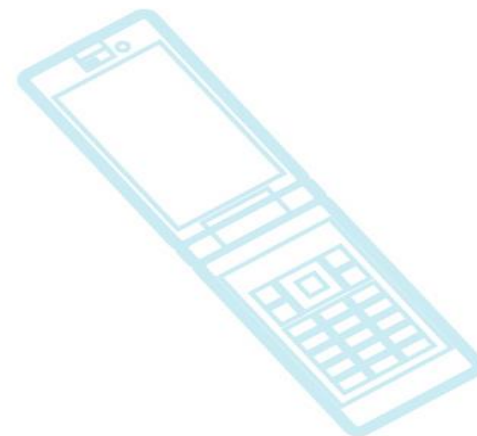
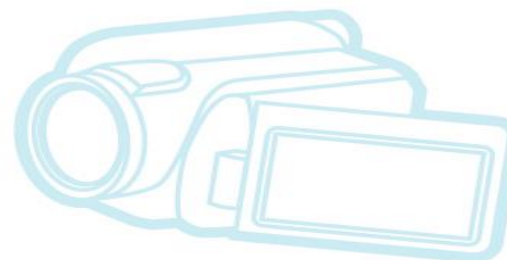
## Linux Kernel

Technology Areas

Industry News

Scorecards

Conclusions





CE Workgroup

# Linux Kernel

- Versions in the last year
  - Pick a few items from each release that are relevant to embedded
    - Lots of things are NOT relevant to embedded (IMHO)
  - Not very good coverage of SoC or driver contributions
- Development Stats





CE Workgroup

# Kernel Versions

- Linux v5.14 – 29 Aug 2021 – 63 days
- Linux v5.15 – 31 Oct 2021 – 63 days
  - “Trick or Treat” release
- Linux v5.16 – 9 Jan 2022 – 70 days
- Linux v5.17 – 20 Mar 2022 – 70 days
- Linux v5.18 – 22 May 2022 – 63 days
- Linux v5.19-rc3 – Recently finished the merge window
  - Expect 5.19 release by end of July



CE Workgroup

# Linux v5.14 (August 2021)

- memfd\_secret system call was added
  - Details on next slide
- new tracers
  - osnoise - show application delays caused by kernel activity
  - timerlat – detailed info about timer-based wakeups
- A fair amount of Qualcomm and MediaTek driver code
  - clocks, pin controllers, sound
- “simplifiedrm” driver
  - direct-rendering interface for simple framebuffer devices



# memfd\_secret system call

- Creates a region of memory that even the kernel cannot directly access
  - Pages are removed from the kernel's direct map
  - Intended to be used for cryptographic info (e.g. keys)
- Makes it difficult for other processes or even the kernel to unintentionally (or even intentionally) access the memory
  - See <https://lwn.net/Articles/835342/>
  - For many more details, see <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=1507f51255c9>



CE Workgroup

# Linux v5.15 (October 2021)

- Realtime preemption locking code – “Sleeping spinlocks”
  - It’s a big deal - More on this later
- Core scheduler support for asymmetric systems
  - Cores on the same chip that can run either 64-bit or 32-bit
    - How to deal with scheduling when the processor can’t even execute some code
  - See <https://lwn.net/Articles/838339/>
- ksmbd – in-kernel SMB server (!!)
  - Not a replacement for Samba, but provides better optimization for Linux in some situations





CE Workgroup

# Linux v5.15 (cont.)

- printk indexing
  - Can extract all printk messages from kernel
    - Is used to detect changes that could break log-parsing tools
- DAMON system merged (Data Access Monitor)
  - See next page
- kernel now uses `-Werror` flag during build, by default
  - A compiler warning will cause the build to fail



CE Workgroup

# DAMON system

- DAMON = Data Access Monitoring tool
- Provides tools to record data access and show visualizations of access patterns
- Different visualizations available
  - A heatmap of memory access for your workload
  - Graphs showing information about working set size
- See <https://damonitor.github.io/doc/html/v17/admin-guide/mm/damon/index.html>
- *Nice diagnostic tool... but is it actually more?*



# Linux v5.16 (January 2022)

- EROFS (Enhanced Read-Only FS) continues to get new features
  - Multiple-device support
- io\_uring operations can now have security policies enforced by SELinux or Smack
- DAMON operation schemes added
  - DAMON can perform pro-active page reclaim, and monitor the physical address space
  - See <https://lwn.net/Articles/863753/>



CE Workgroup

# Linux v5.17 (March 2022)

- Kernel can decompress kernel modules within itself (instead of relying on user space)
  - This helps the LoadPin security module
- RTLA – realtime analysis tools have been added
  - osnoise and timerlat
- Some changes to flags fields used in FUSE\_INIT call
  - Check your FUSE filesystems and tools for compatibility





CE Workgroup

# Linux v5.18 (May 2022)

- Support for older ARM (ARMv4, ARMv5) MMU-less systems has been removed.
  - Support for MMU-less ARMv7-M remains
- Lots of RISC-V stuff
- Tracing system supports “user events”, to allow dynamic tracepoints in user-space applications
- Kernel compiles against C11 language standard (instead of C89)
  - See <https://lwn.net/Articles/885941/>



CE Workgroup

# Linux v5.19-rc3 (in progress)

- More SoC support and device drivers
- ARM multi-platform work completed (mostly)
- Initial support for LoongArch CPU architecture
  - New RISC instruction set architecture - similar to MIPS or RISC-V
  - By Loongson (Chinese chip manufacturer)
- New hardware timestamp engine subsystem
  - Devices that can record timestamps on hardware triggers



# Linux 5.18 developer stats

- 14,954 change sets, by 2024 developers (289 new devs.)
- Most active 5.18 developers, by changesets:

Person	Changesets	Percent	Subsystem Area
Krzysztof Kozlowski	214	1.4%	device tree updates
Matthew Wilcox	164	1.1%	folio patches
Christoph Hellwig	154	1.0%	refactoring of block and fs layers
Geert Uytterhoeven	140	0.9%	Renesas pin control
Ville Syrjälä	135	0.9%	i915 graphics driver



# Linux 5.18 developer stats

- Most active 5.18 developers
- By lines of code:

Person	Lines changed	Percent	Subsystem area
Leo Li	227676	19.4%	AMD graphics driver
Quinquin Zhuo	197757	16.9%	AMD graphics driver
Ian Rogers	72008	6.1%	perf tool
Alan Kao	15814	1.3%	removed nds32 architecture
Ming Qian	12176	1.0%	Amphion media drivers





# Kernel commit log entries

- Number of commit log entries (including merges), per kernel version

Company	git log count	developer count
5.14	15871	1982
5.15	13473	1853
5.16	15385	2074
5.17	14199	1978
5.18	16205	2116
5.19-rc1	13973*	1846*

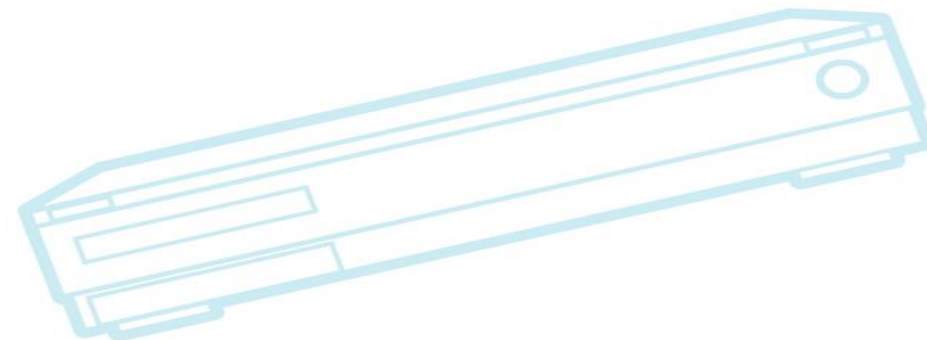
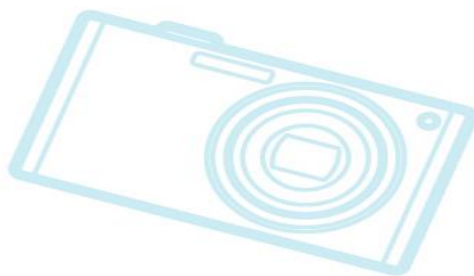
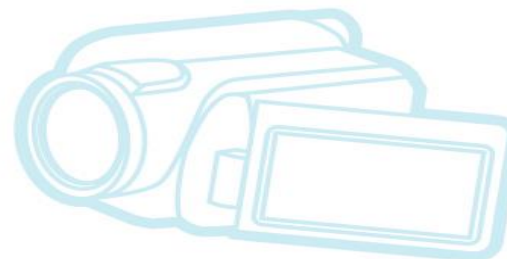
- \*we're not done with the 5.19 release cycle yet
- Extracted using 'git log v5.yy..v5.zz --oneline | wc -l' and 'author-stats v5.yy..v5.zz | wc -l'



CE Workgroup

# More to add...

- Get stats for bug reports
  - 40% are by automated testing robots

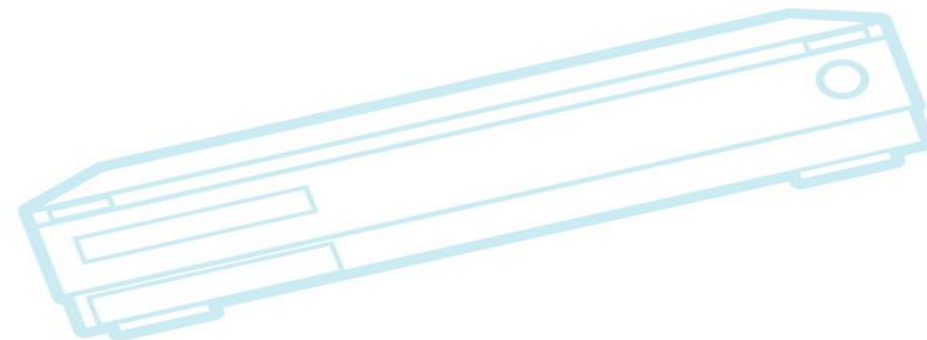
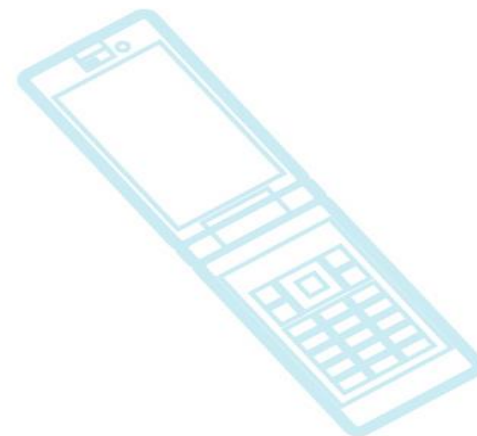
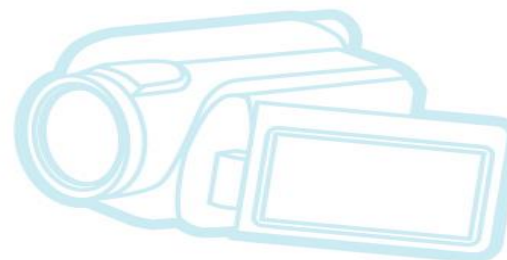
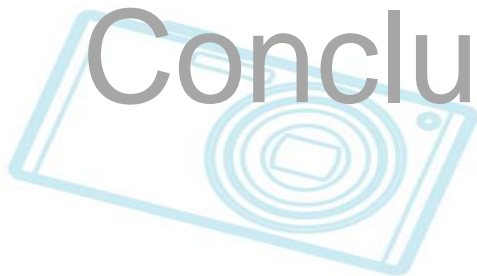




CE Workgroup

# Outline

Linux Kernel  
**Technology Areas**  
Industry News  
Scorecards  
Conclusions





CE Workgroup

# Technology Areas

- Audio
- Core Kernel
- Filesystems
- Graphics
- Networking
- Security
- Testing
- Toolchains
- Tracing





CE Workgroup

# Audio

- PipeWire continues to gain ground
  - Replaces PulseAudio and JACK
  - Has higher performance
  - WirePlumber is a new session manager for PipeWire
    - Scriptable in LUA
  - See Talk by Geoge Kiagiadakis (Collabora) from ELC 2021
    - [https://elinux.org/images/f/fb/Master\\_your\\_pipewire\\_streams\\_with\\_wireplumber.pdf](https://elinux.org/images/f/fb/Master_your_pipewire_streams_with_wireplumber.pdf)
- PulseAudio 16.0 Released in May
  - Lots of fixes and improvements
    - One item: pactl can output info in JSON
  - See <https://www.freedesktop.org/wiki/Software/PulseAudio/Notes/16.0/>



CE Workgroup

# Core Kernel

- memfd\_secret (v5.14)
- printk indexing (v5.15)
- scheduling for asymmetric processors (v5.15)
- Kernel compiles against C11 spec. (v5.18)
  - Mostly has to do with where variables can be declared
    - Want to declare iterator variable in loop itself, so macros like 'list\_for\_each\_entry' can avoid leaking the iterator value
      - Which can result in a speculative execution vulnerability
  - See <https://lwn.net/Articles/886516/>



# Rust in the Linux kernel

- Work to support Rust code in the Linux kernel continues
  - Patches are still considered experimental
- Support moved to version 1.59.0 of the Rust language
- Patch series is about 35,000 lines of code
- Have been recent discussions about what library code from Rust might be useful in kernel context
  - There is no apparent plan to support auto-loading of Rust code from Rust's dynamic package repository (crates.io)
- Most kernel developers still in “wait-and-see” mode
- See <https://lwn.net/Articles/889924>





# Filesystems and I/O

- io\_uring continues to mature
  - As a reminder: see <https://lwn.net/Articles/810414/>
  - Performance enhancements (v5.15)
  - Can be security-regulated by SELinux or Smack (v5.16)
  - Support for zero-copy networking (coming)
- EROFS and F2FS continue to mature
  - Better compression support
  - Better xattr support
- FUSE\_INIT flag changes (v5.17)





CE Workgroup

# Graphics

- “simplifiedrm” driver merged in 5.14
- legacy fbdev sub-system got a new maintainer (Jan, 2022)
  - But there was some initial friction
  - See <https://lwn.net/Articles/881827/>
- MALI GPUS now have an fully-conformant OpenGL ES 3.1 (Panfrost) driver
  - Supports the new Valhall GPU architecture
  - Patches are queued for upstream
  - See <https://www.collabora.com/news-and-blog/news-and-events/conformant-open-source-support-for-mali-g57.html>



# Graphics – NVIDIA GPU Code

- NVIDIA transitioning to open source driver!
  - Marks a big shift in open source policy by company
  - Code is not upstream yet
    - Kernel portion of GPU driver is available, but ABI/API not stabilized yet
  - Some proprietary code still in user-space
    - e.g. OpenGL / Vulkan / OpenCL / CUDA drivers
  - Published driver code is under dual MIT/GPLv2 license
  - Not sure effect on Nouveau driver yet
    - They can use the published code
- See <https://www.phoronix.com/scan.php?page=article&item=nvidia-open-kernel&num=1>



CE Workgroup

# Networking

- Always a stream of oddball networking features and enhancements:
  - Custom configuration of hash policies for multipath IP traffic
  - Support for Management Component Transport Protocol (MCTP)
  - Unix-domain sockets now support out-of-band data
  - `SO_RESERVE_MEM` can reserve kernel memory and speed up some operations
  - New `sysctl` knobs for tuning the ARP cache behavior
  - And so on...
- New internal function to provide reason for a packet drop (5.17)
  - Helps administrators determine reason for networking issues
    - But only 63 of 4000 `kfree_skb` calls converted so far
  - See <https://lwn.net/Articles/885729/>



CE Workgroup

# Real-Time

- **rtla – real-time Linux analysis tool (5.17)**
  - See [https://www.phoronix.com/scan.php?page=news\\_item&px=Linux-5.17-RTLA](https://www.phoronix.com/scan.php?page=news_item&px=Linux-5.17-RTLA)
- **PREEMPT\_RT status**
  - Sleeping locks was mainlined (v5.16)
  - Patches have been going in continuously – through 5.19





# RT preemption locking code

- Provides “sleeping spinlocks” (and sleeping rwlocks)
  - Allows for process switch (schedule) while a lock is held, which is the core feature of PREEMPT\_RT
- Must turn on CONFIG\_PREEMPT\_RT config option
  - Extensively tested to verify that non-RT kernels are not affected
  - See also CONFIG\_RT\_MUTEXES
- See the commit for details:  
<https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=e5e726f7bb9f>
- Merged in kernel v5.15
  - After 17 years of development effort and many reworks and refactorings



# PREEMPT\_RT - What's left

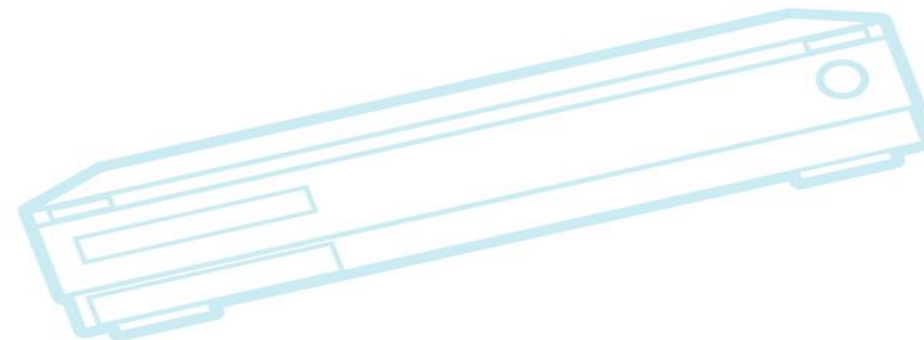
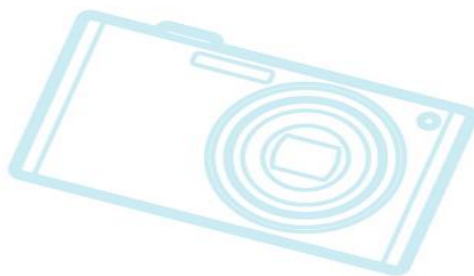
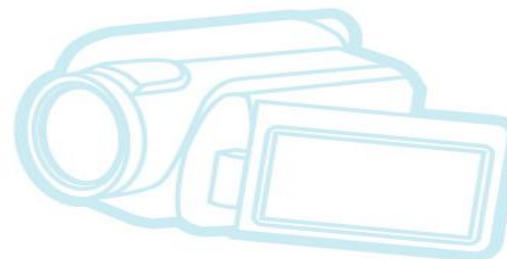
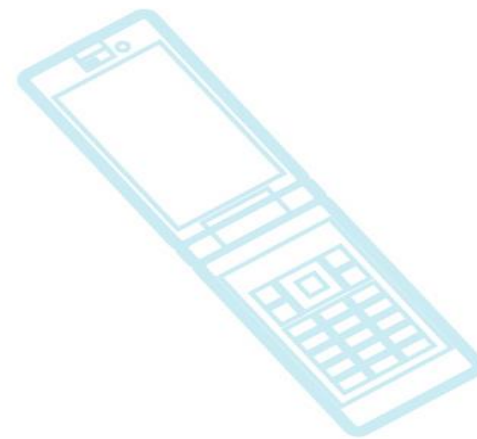
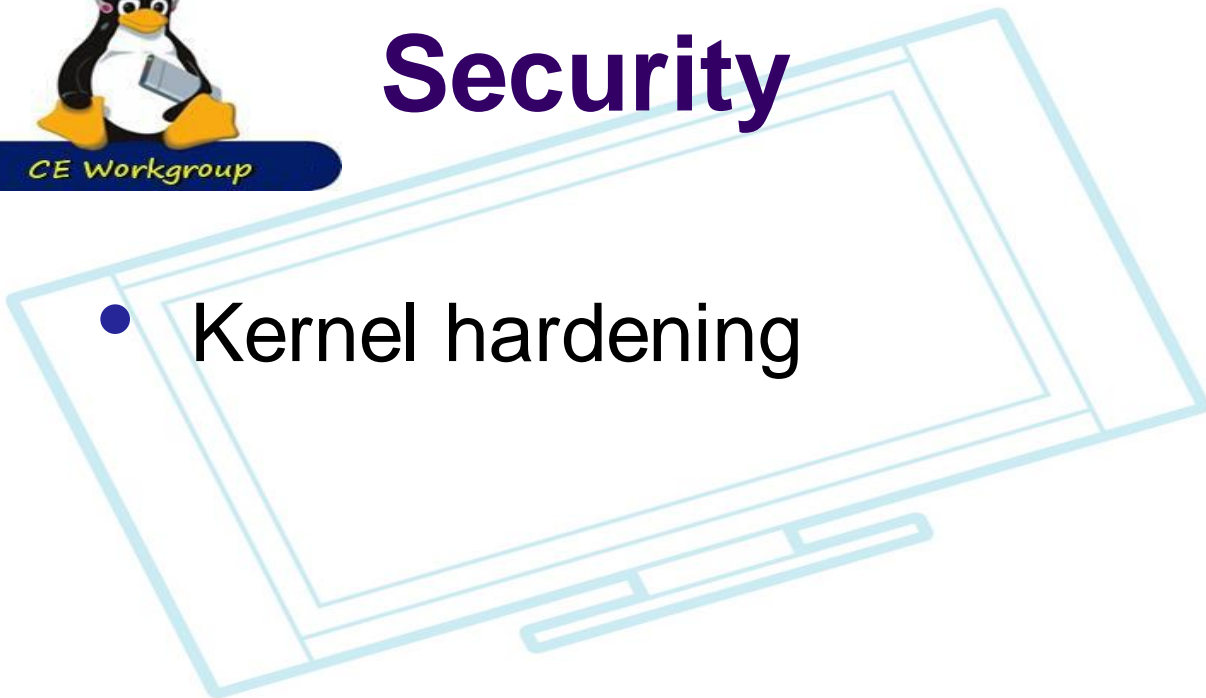
- What's left in PREEMPT\_RT patches out of mainline:
  - About 1300 lines of code, affecting 92 files (in 51 patches) (!!)
    - Improved by about 1700 lines of code, 40 files and 50 patches since Feb.
    - Some big changes to printk
      - But I heard they got backed out...
    - Some changes to the zram driver, 8250 serial driver, and the core scheduler (and other places)
  - People are anxious for Linux RT without having to apply a patch
- See <https://mirrors.edge.kernel.org/pub/linux/kernel/projects/rt/5.19/patches-5.19-rc1-rt1.tar.gz>
- See the RT BOF on Friday



CE Workgroup

# Security

- Kernel hardening





# Kernel hardening

- Control flow integrity (CFI) (v5.13)
- strict memcpy() bounds checking (v5.16)
  - <https://lwn.net/Articles/864521/>
- Spectre mitigations
  - There always seem to be new speculative execution vulnerability mitigations
  - Interestingly, v5.16 removed some Spectre-mitigation behavior for seccomp()
    - Devs decided that the extra mitigations weren't really buying more security

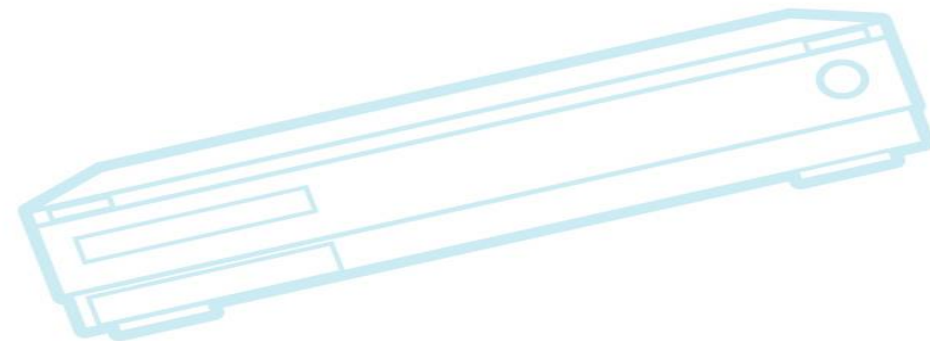
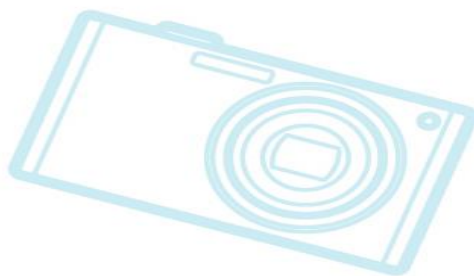
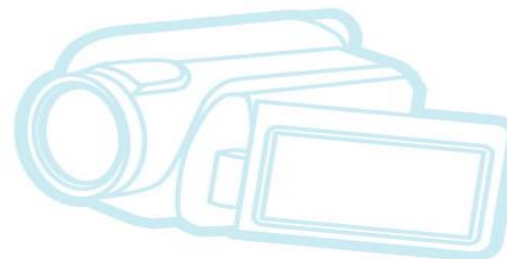
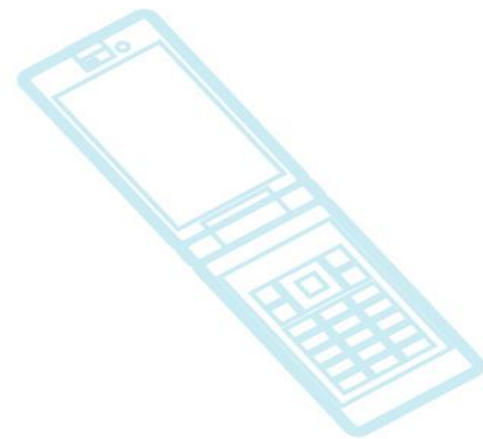




CE Workgroup

# Testing

- Systems
- Suites of tests





CE Workgroup

# Test Systems

- 0-Day – reports bugs at time of patch submission
- KernelCI – has added kselftest git repo to list of trees it tests
- Syzbot – always producing more fuzzing failure cases
- CKI – providing many reports to upstream
- LKFT – providing many reports to upstream
- CompassCI
  - watch this one!
    - Is by the same developer who wrote 0-Day for Intel (Wu Fengguang - who is now at Huawei)
  - See <https://static.linaro.org/connect/lvc21/presentations/lvc21-202.pdf>



CE Workgroup

# Test Suites

- LTP
  - Latest release: 20220527 (May 27, 2022)
    - New test and fixes to tests
    - Added 'test max runtime' concept to replace test timeout
      - <https://people.kernel.org/metan/test-timeout-and-runtime>
    - See <https://github.com/linux-test-project/ltptest/releases>
- kselftest
  - New tests for arm64 (fp, signal), BPF, network drivers, kvm, network forwarding, netfilter, powerpc, ftrace user\_events, vm
  - Patch being discussed to taint kernel if testing module is loaded
- Kunit
  - Continues to grow (old tests converted over to Kunit)



CE Workgroup

# Toolchains - GCC

- Kernel now requires gcc 5.1 to build
- GCC 12.1 released May 6, 2022
  - GCC can now initialize all stack variables implicitly
    - Intended to eliminate flaws related to uninitialized stack variables
    - Use ‘`-ftrivial-auto-var-init=pattern`’
      - Can fill variables with repeated 0xFE pattern, which tends to show bugs
      - Can fill variables with 0, which tends to provide safer state
  - The experimental static analyzer now has uninitialized variable use detection
  - See <https://gcc.gnu.org/gcc-12/changes.html>





# Toolchains - LLVM

- LLVM 14.0.4 released May 24, 2022
  - See <https://releases.lldm.org/14.0.0/docs/ReleaseNotes.html>
- People are using for whole distributions, not just kernel
- See presentation: “Experiences of OS distributions using LLVM as their main toolchain”
  - By Bernhard Rosenkranzer at 2022 European LLVM Dev. meeting
  - <https://youtu.be/h9xg8Y8bylg>



# Tracing

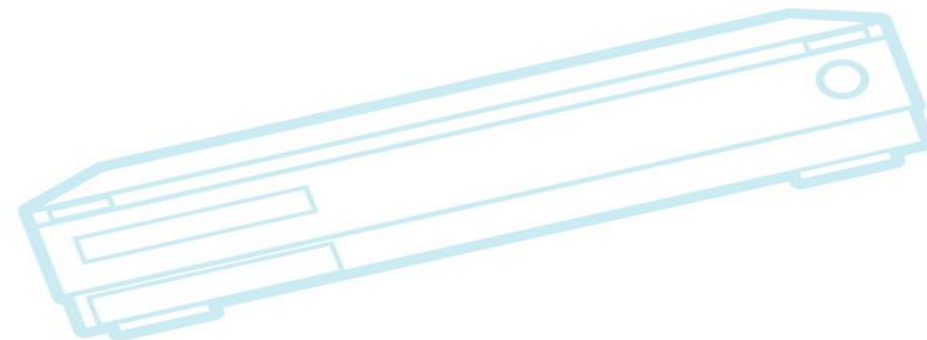
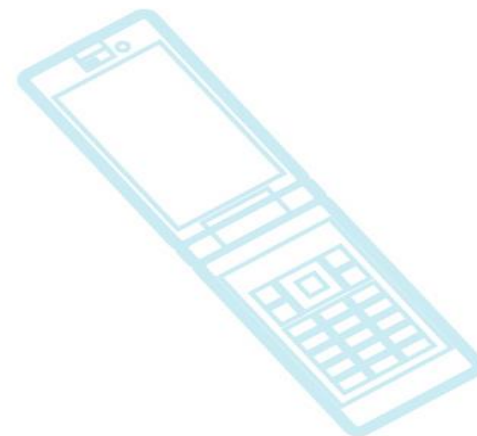
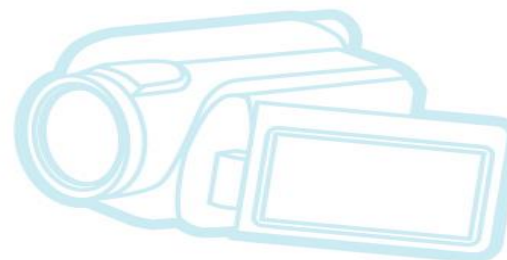
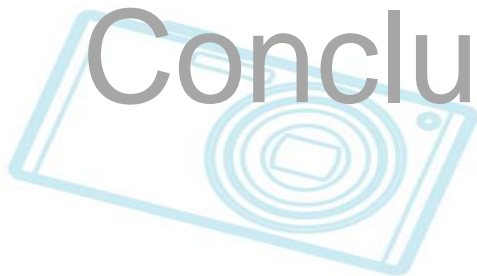
- New perf features in v5.18:
  - <https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=7b58b82b86c8>
- SystemTap 4.17 released
  - tracing toolkit for Linux kernel
  - See <https://lwn.net/Articles/893682/>
- Hardware Timestamping Engine system added (v5.18)
  - Can have hardware grab timestamp automatically on a hw event
    - Good for low-overhead tracing
  - See <https://docs.kernel.org/hte/hte.html>



CE Workgroup

# Outline

Linux Kernel  
Technology Areas  
**Industry News**  
Scorecards  
Conclusions

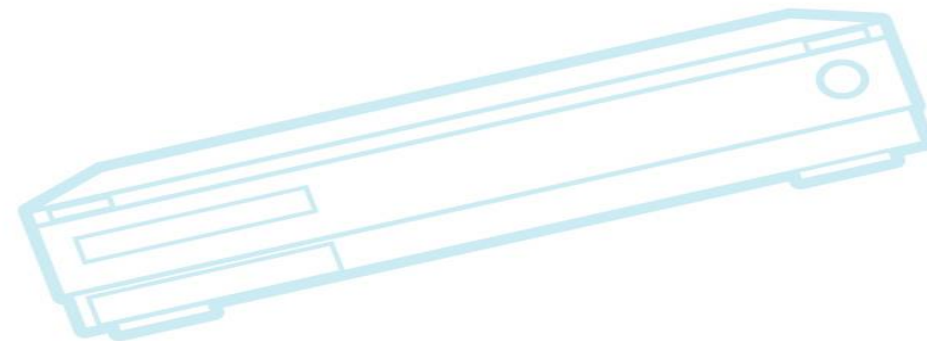
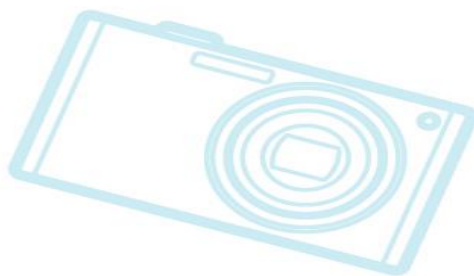
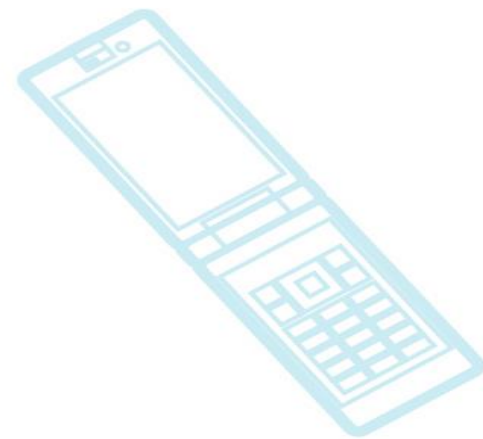




CE Workgroup

# Industry News

- Open Source Security Foundation
  - Alpha-Omega project
- Miscellaneous
  - Interesting cases of embedded Linux







CE Workgroup

# OpenSSF

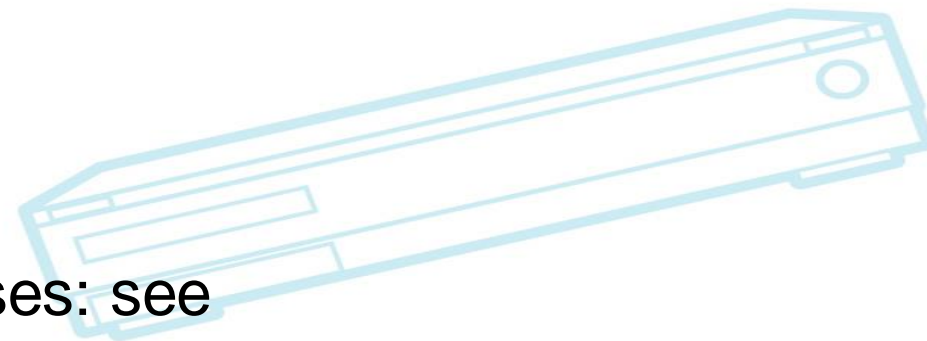
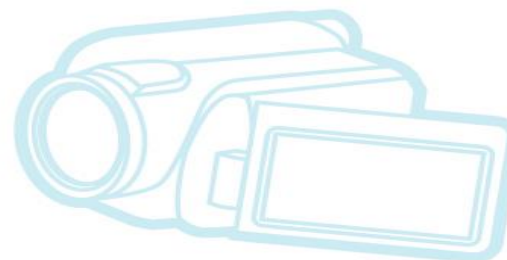
- Open Source Security Foundation
- Comprehensive project to enhance OSS security
  - Vulnerability disclosures
  - Security tooling
  - Best practices and training
  - Securing critical projects
- Well-funded and well-connected
  - Represented OSS at white house security summit and at congressional hearings



CE Workgroup

# OpenSSF Activities

- Initiatives:
  - Security Scorecard
  - Security Reviews
  - Security Metrics Dashboard
  - Package Feeds / Package Analysis
  - CII Best Practices Badge Program
- Standards:
  - Open Source Vulnerability Schema
  - Supply-Chain Levels for Software Artifacts
- Guides and Training:
  - OSS Vulnerability Guide
  - Free Security Software Development courses: see <https://openssf.org/training/courses/>





CE Workgroup

# Alpha-Omega Project

- Effort to systematically search for vulnerabilities in open source code
- Alpha = highest priority (most-used, most-critical) projects
  - Will directly assist projects with security (ie with developer resources)
- Omega = long tail of open source projects (10,000 projects)
  - Plan to provide training, and apply automated security analysis, scoring and remediation
- See <https://openssf.org/community/alpha-omega/>



# Miscellaneous

- Intel acquired Linutronix
  - Linutronix is Thomas Gleixner's Linux company
    - Represents support for realtime efforts by Thomas?
  - See <https://community.intel.com/t5/Blogs/Products-and-Solutions/Software/Intel-Acquires-Linutronix/post/1362692>
- Oniro – (New?) Eclipse Foundation IoT operation system
  - Distributed IoT OS??
  - Can sit on top of Linux, Zephyr, FreeRTOS or LiteOS kernel
  - Has blueprints to build end-user ready products (e.g. vending machine)
  - Yocto-based
    - Builds entire system at one time

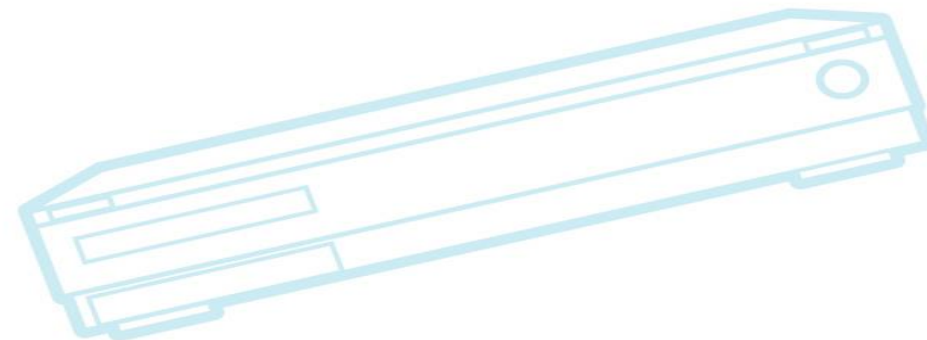
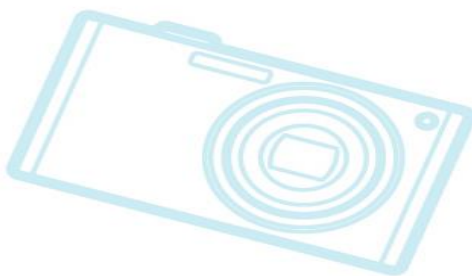




CE Workgroup

# Interesting embedded Linux uses

- Mars Ingenuity helicopter





CE Workgroup

# Mars Helicopter - Ingenuity





CE Workgroup

# Mars Helicopter

- Mars Ingenuity Helicopter landed in February, 2021 on Mars
- Performed tests and demonstrations in April & May (2021)
  - First 5 flights were part of “Technology Demonstration”
- After demo, NASA created a plan for continued flights
- Is still flying...
  - Has performed 29 flights so far
- Updates:
  - Recent flight to lander backshell
  - Recent hardware/software issues







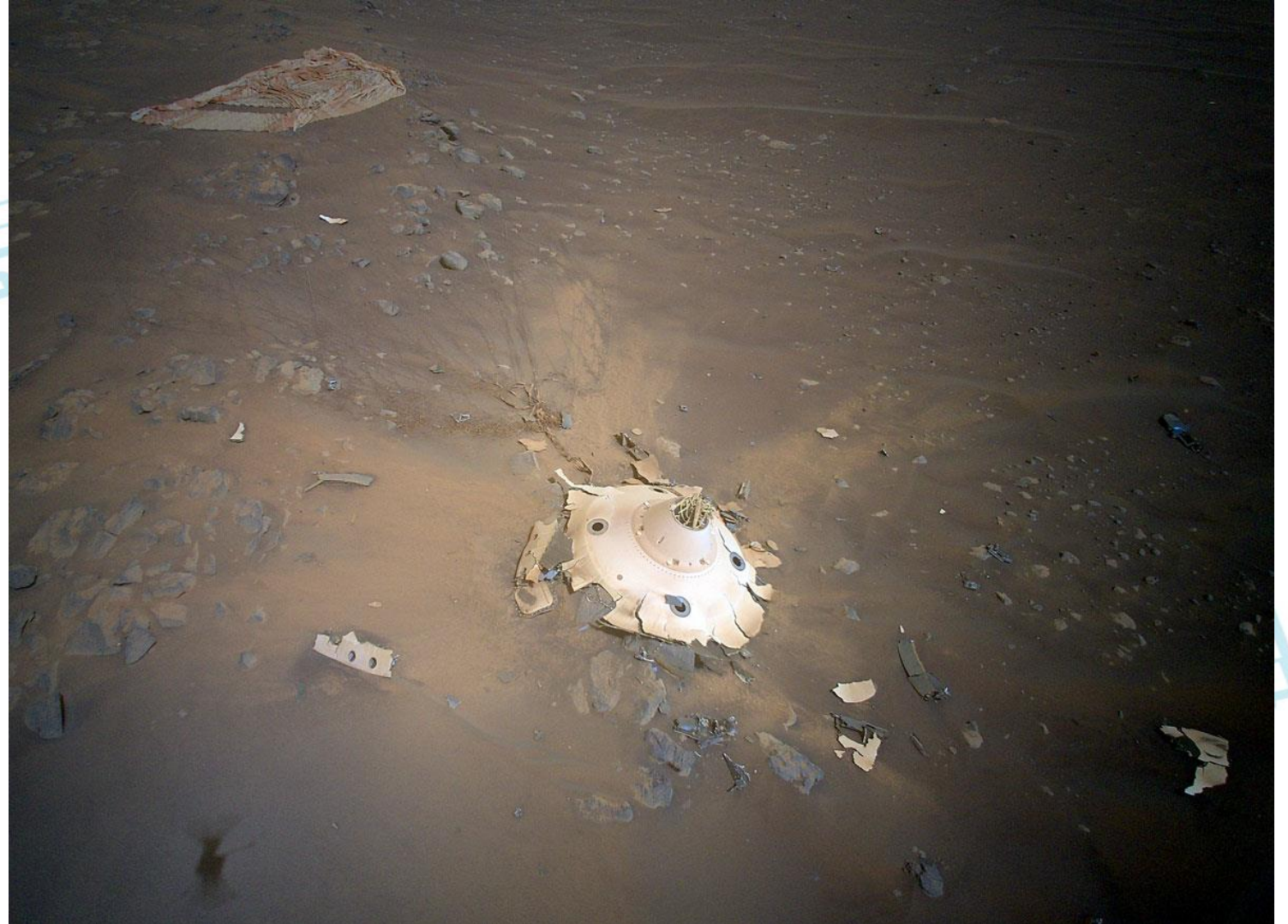
CE Workgroup

# Picture of Lander backshell (crash site)

The Rover and Helicopter were near the Lander backshell crash site in April.

The helicopter was diverted to take pictures of the area. (Flight 26)

This is the first time that crash debris has been recorded in such detail (from low air vantage point) on another planet



Source: <https://mars.nasa.gov/resources/26694/rovers-backshell-seen-from-the-air/>





CE Workgroup

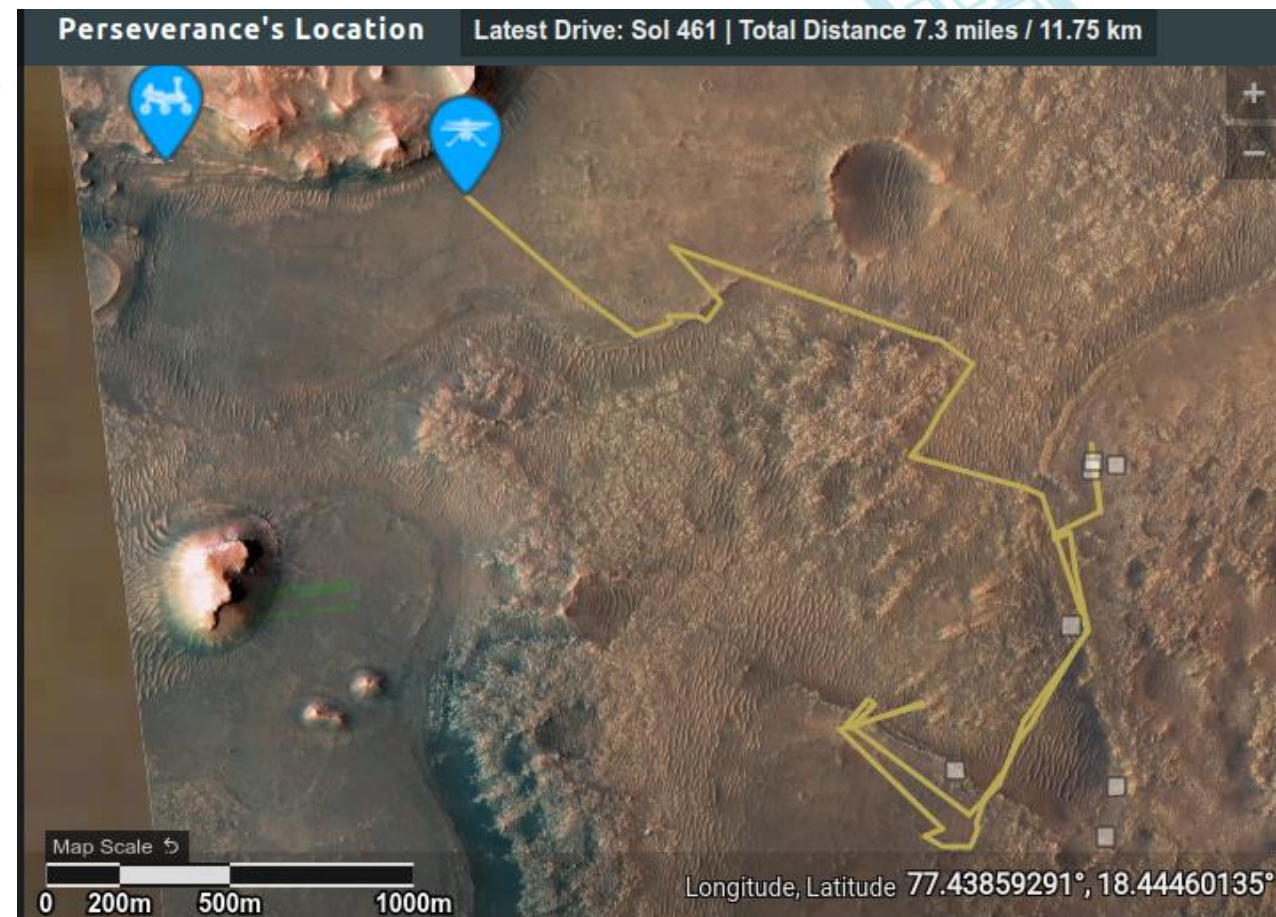
# Ingenuity Helicopter Update (June 2022)

- Recent hardware/software issues:
  - Colder weather, and dust, has required changes to thermal management
    - Dust has reduced solar panel charging
    - Helicopter shuts down at night, and allows lower core temperature
  - Loss of communications (due to a clock reset due to low charge)
  - Inclinometer is now broken
    - NASA is sending a patch to use information from other sensors in place of inclinometer data
      - Will perform a software update on Mars!!
      - Note: patch was already written!
  - See <https://phys.org/news/2022-06-nasa-mars-helicopter-ingenuity-patch.html> and <https://mars.nasa.gov/technology/helicopter/status/385/keeping-our-sense-of-direction-dealing-with-a-dead-sensor/>



CE Workgroup

# Ingenuity flights on Mars







# Sources for Mars helicopter

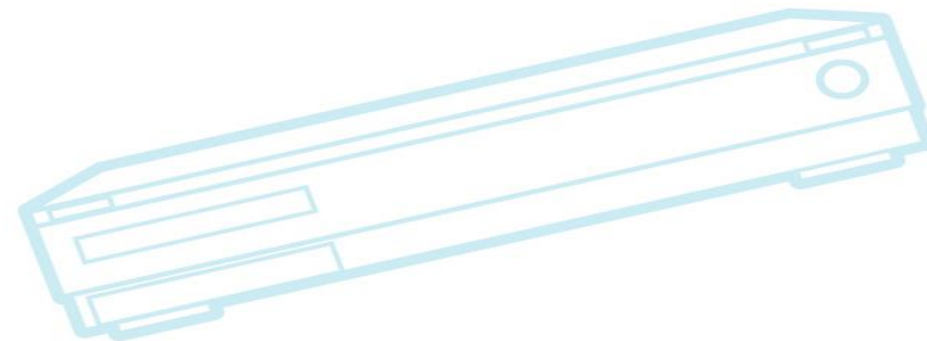
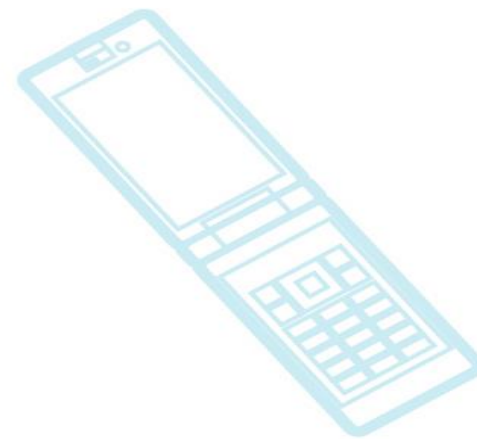
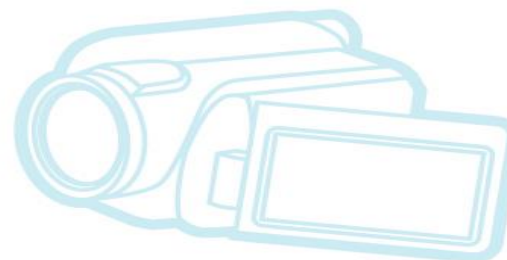
- Talk by Tim Canham at ELC 2021
  - Slides: [https://elinux.org/images/5/5a/1.\\_TIMOTHY\\_CANHAM.pdf](https://elinux.org/images/5/5a/1._TIMOTHY_CANHAM.pdf)
  - Video: [https://youtu.be/0\\_GfMcBmbCg](https://youtu.be/0_GfMcBmbCg)
- <https://mars.nasa.gov/technology/helicopter/>
- [https://en.wikipedia.org/wiki/Ingenuity\\_\(helicopter\)](https://en.wikipedia.org/wiki/Ingenuity_(helicopter))
- <https://thenewstack.io/how-the-first-helicopter-on-mars-uses-off-the-shelf-hardware-and-linux/>
- <https://www.pcmag.com/news/4-android-smartphones-with-as-much-power-as-nasas-mars-helicopter>



CE Workgroup

# Outline

Linux Kernel  
Technology Areas  
Industry News  
**Scorecards**  
Resources



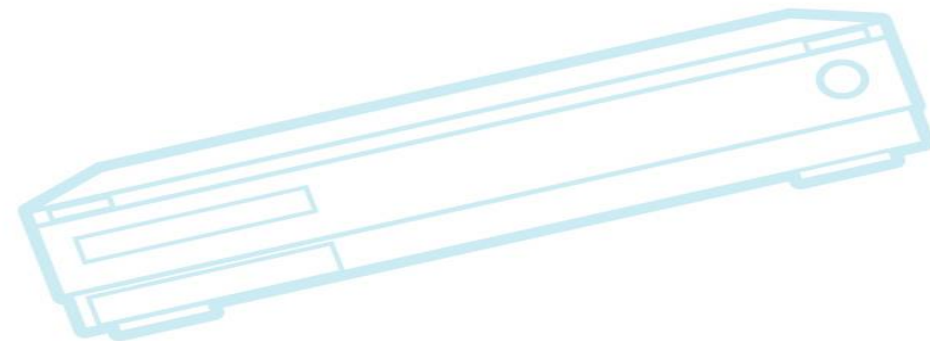
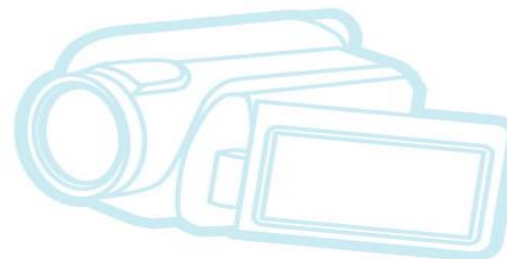
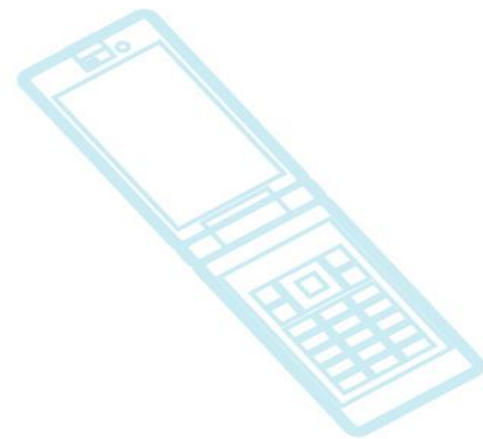




CE Workgroup

# Scorecards

- Are we done yet?
- Well, where are we?
- Scorecards for:
  - Technology
  - Development
  - Markets

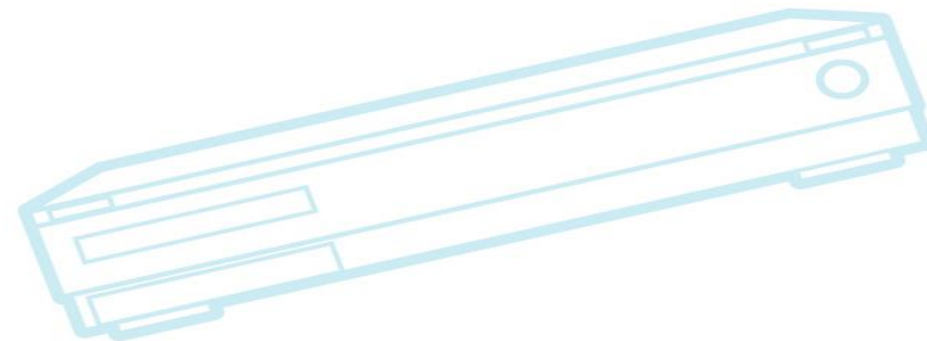
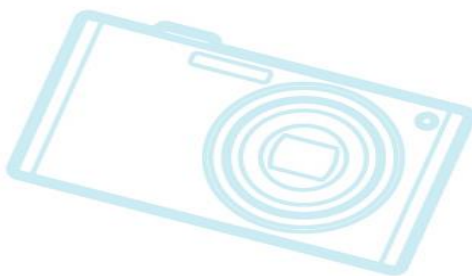
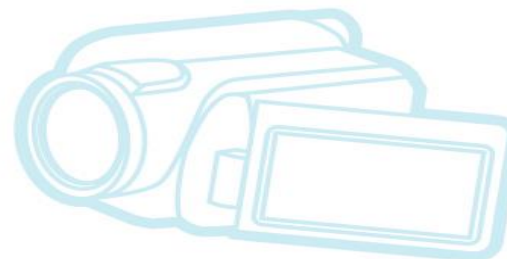




CE Workgroup

# Technology Scorecard

- Original focus areas:
  - System size
  - Boot time
  - Power management
  - Realtime
  - Security





CE Workgroup

# Technology Scorecard

- Based on contributions in the last few years:
- System size – done
- Boot time – done
- Power management – done
- Realtime – done
- Security – in progress
- Congratulations everyone!! We did it!!
  - World(s) domination achieved!



# Technology Scorecard (reality and explanations)

- System size – **done**
  - Lower limit is about 16M, and it's not getting any better
  - Alas, Linux will never be on 1-cent processors
    - The 10 trillion IoT sensors will be running something else (darn)
- Boot time – **done**
  - Cold boot boot time reduction largely abandoned in mainline
    - People do heroic special-casing to get low boot times, when required
  - Most products use either suspend/resume or low-power idle





CE Workgroup

# Technology Scorecard (explanations)

- Power Management – **done??**
  - governors, frequency scaling, power domains, power qos, and power scheduling features are all upstream
  - It requires SoC and board support (e.g. driver pm integration) for it to work
  - It's now mostly a BSP (Board Support Package) problem



CE Workgroup

# Technology Scorecard (explanations)

- Realtime – **done??**
  - **PREEMPT\_RT** code is (*almost all*) upstream!!
  - But it requires ongoing maintenance to avoid changes that damage realtime performance
- Security – **in progress**
  - kernel hardening, handling security reports, Rust drivers
  - Alpha/Omega project



CE Workgroup

# Real Technology Scorecard

- System size – done
- Boot time – done
- Power management – done??
- Realtime – done??
- Security – in progress

Recognise now that all of these are “holistic”.

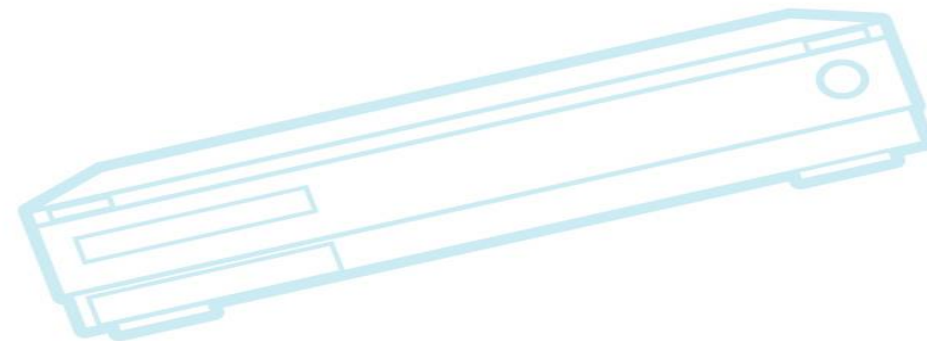
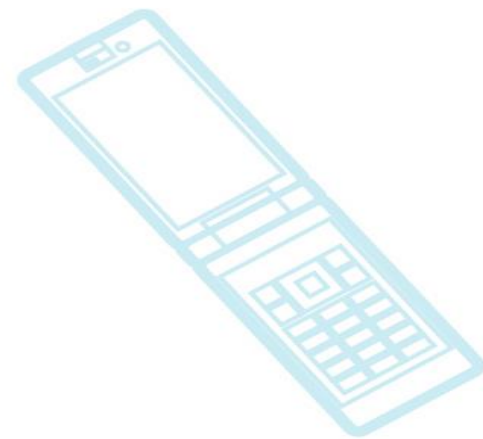
- All of these require pervasive, constant maintenance, distributed throughout the code base



CE Workgroup

# Development Scorecard

- Build systems/Distros
  - Training/Consulting
  - Toolchains
  - Debugging capabilities
  - Test Systems
  - Hardware support
- options: good, in progress, lacking







CE Workgroup

# Development Scorecard

- Build systems/Distros - **good**
  - Yocto Project, Buildroot, Debian, and specialized ones: OpenWrt, Android
  - Still being worked on, but are mature and featurefull
- Training/Consulting – **good**
  - There have never been more books, tutorials, training
  - Lots of companies and resources available to help build products
    - Pengutronix, Linutronix, Bootlin, Collabora, PathPartner, Mentor, Wind River, Mender, Witekio, Konsulko, Montavista, Timesys, BayLibre
- Toolchains – **good**
  - both gcc and llvm are useful (including for cross-development)
  - SoC vendors (and others) add support for new instructions sets



CE Workgroup

# Development Scorecard

- Debugging capabilities – good
  - Lots of options for tracing, debugging, diagnostics
- Test Systems – in progress
  - CI systems and test suites are available
    - But automated test coverage can be improved
  - Automated testing still has gaps (particularly hardware testing)



CE Workgroup

# Development Scorecard

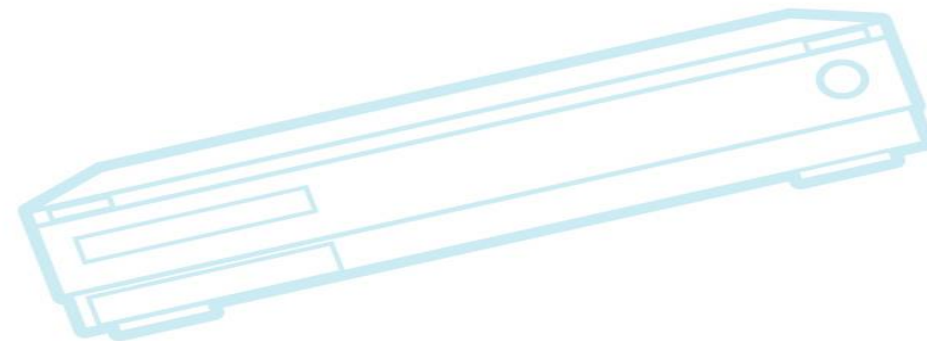
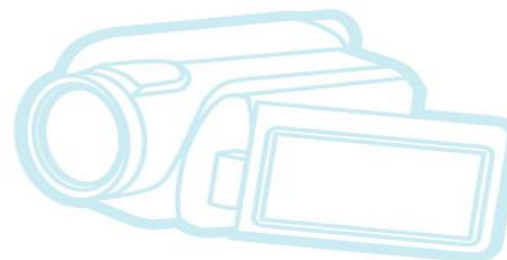
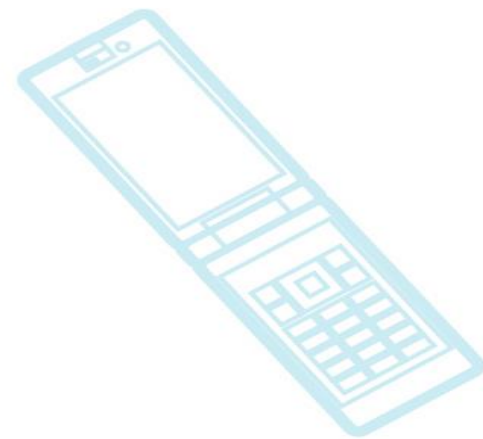
- Hardware support – in progress
  - SoC vendors provide support for their chips: Intel, ARM (Linaro), RISC-V, MIPS, etc.
    - Not all of it is upstream
  - Sometimes, it's still hard to get vendors to mainline their hardware support
  - Requires driver work by product makers, when drivers should already be in mainline
    - Product makers carry too much technical debt (ie out-of-tree patches)



CE Workgroup

# Development Scorecard

- Build systems/Distros - good
- Training/Consulting - good
- Toolchains – good
- Debugging capabilities – good
- Languages – good
- Test Systems – in progress
- Hardware support – in progress







CE Workgroup

# Markets Scorecard

- Drones
- Robots
- Cars
- Space systems
- Routers
- Mobile Phones
- Consumer electronics (TVs, DVRs, Cameras)
- *I know this list is not comprehensive*



CE Workgroup

# Verticals Scorecard

- Drones - good
  - Walmart just announced expansion of drone deliveries to 34 states
  - Amazon announced delivery trials in California by end of year
  - Lots of commercial drones running Linux
  - DroneCode project
- Robots – good
  - Robot Operating Systems (ROS) and ROS2
  - Projection that at least 55% of total commercial robots shipped in 2024 will have at least one ROS package
    - See <https://www.businesswire.com/news/home/20190516005135/en/Rise-ROS-55-total-commercial-robots-shipped>
- Cars – good
  - AGL just released UCB 13 in April – instrument cluster and infotainment
  - Tesla self-driving uses Linux
  - Lots of Linux in telematics



# Verticals Scorecard

- Space Systems - improving
  - Starlink, SpaceX rockets, Mars helicopter, some cubesats use Linux
  - Commodity hardware in space will probably use Linux more often
- Routers – good
  - Anyone can build a router with Linux these days
- Mobile Phones – good
  - Android market share is 70%
- Consumer electronics (TVs, DVRs, Cameras, Audio) - good
  - In many segments, Linux has almost 100% market share



CE Workgroup

# Markets Scorecard

- Drones - good
- Robots – good
- Cars – good
- Space systems – improving
- Routers - good
- Consumer electronics (TVs, DVRs, Cameras) - good
- Mobile Phones – good

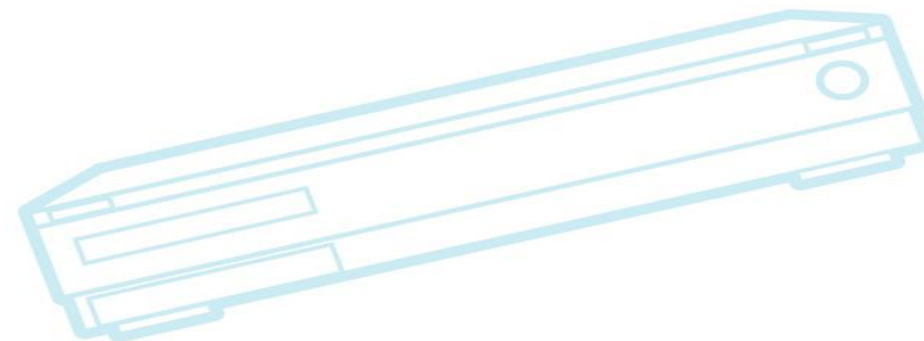
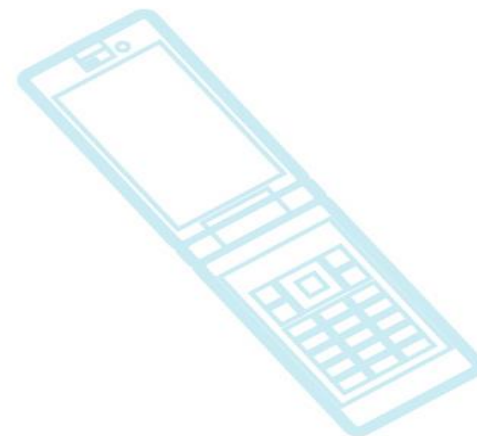
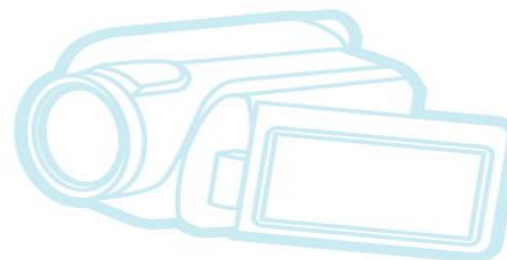
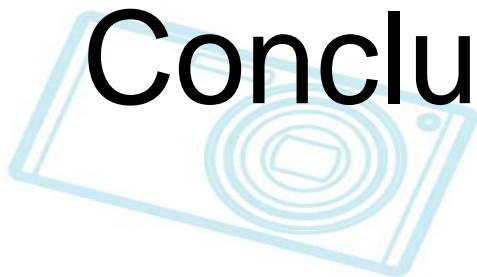




CE Workgroup

# Outline

Linux Kernel  
Technology Areas  
Industry News  
Scorecards  
**Conclusions**





CE Workgroup

# Conclusions

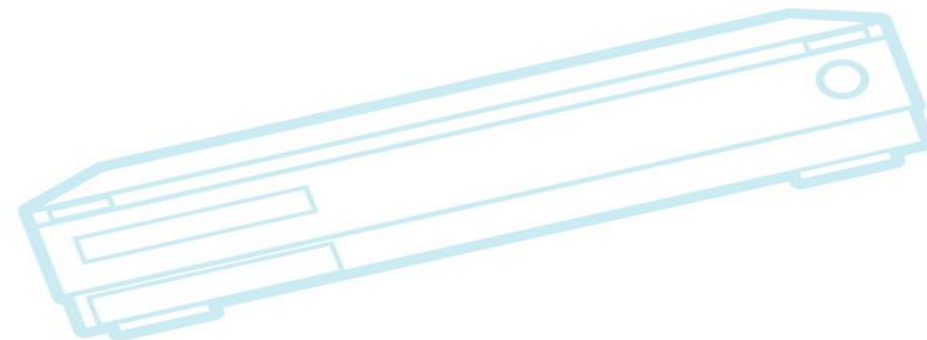
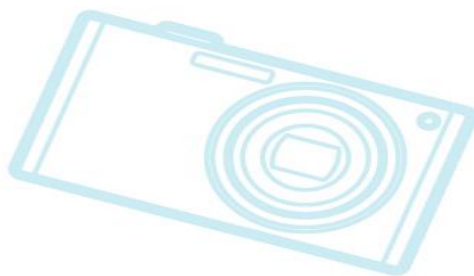
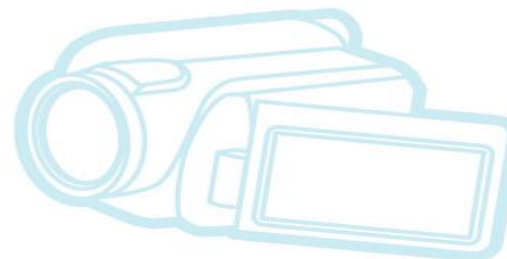
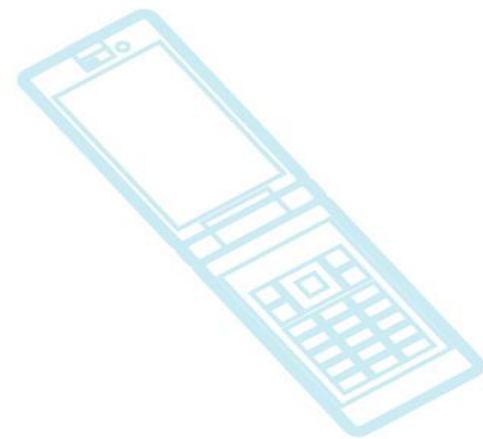
- Overall – we're doing pretty well
  - Widely deployed and functional (billions of devices)
- Linux will not be in low-end, low-power IoT devices
  - e.g. 1M RAM device running on harvested energy
  - I don't expect to see Linux running on a cereal box any time soon
- Core kernel systems are in place to support embedded
  - But new hardware keeps being made
  - We'll always have things to write and upstream
  - Core code can always be improved



CE Workgroup

# Conclusions

- We're not done yet!

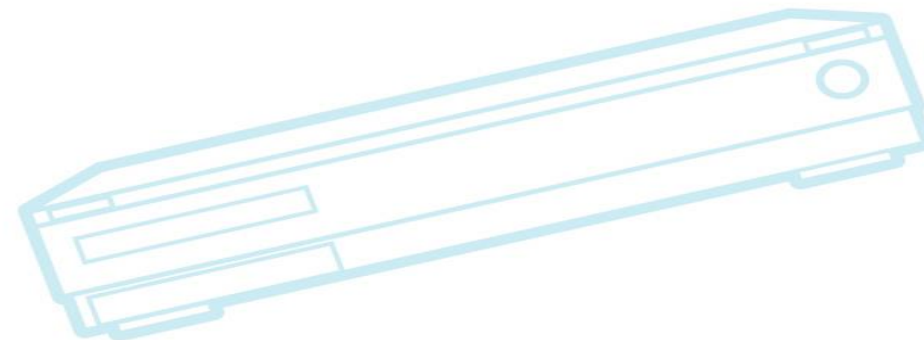
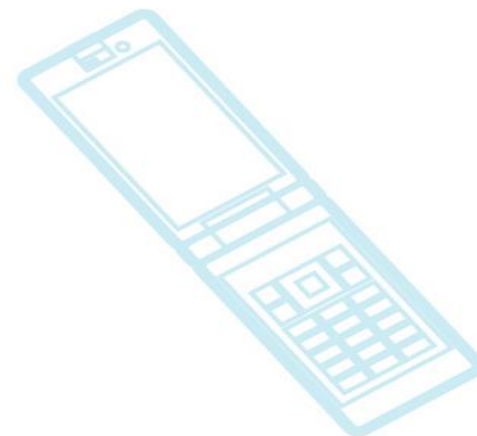
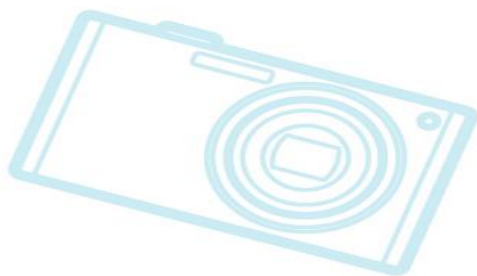




CE Workgroup

# Conclusions

- We're not done yet!
- That's OK – we've got job security for the foreseeable future







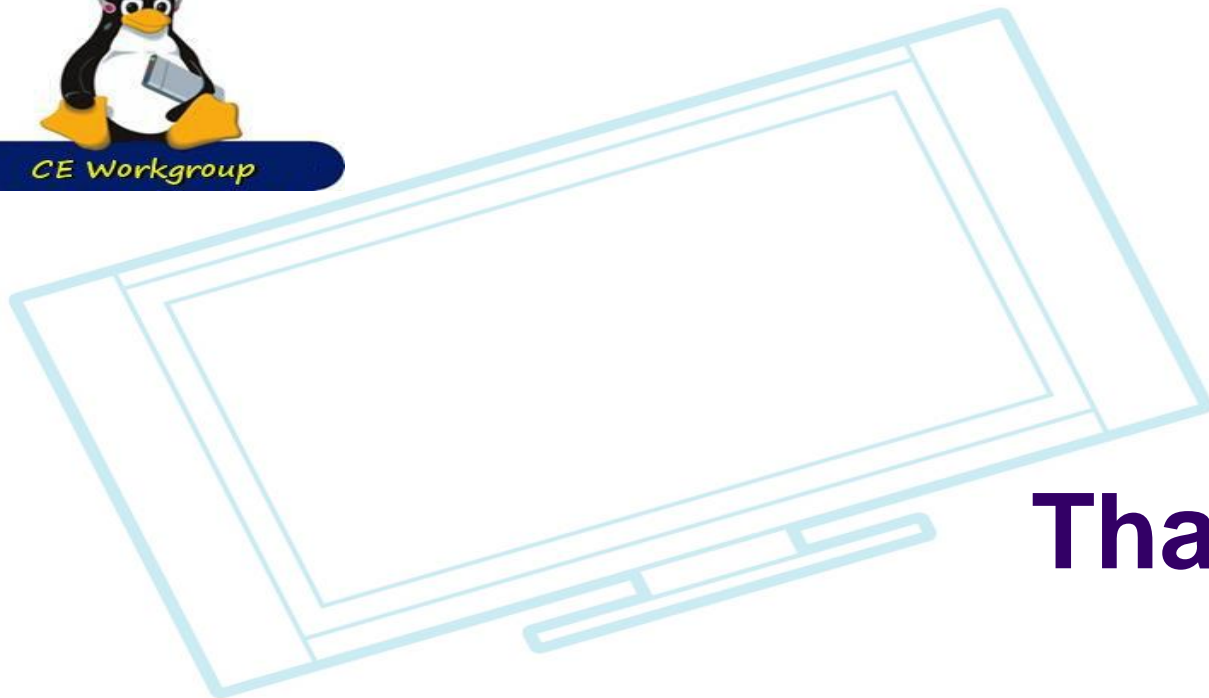
CE Workgroup

# Conclusions

- We're not done yet!
- That's OK – we've got job security for the foreseeable future
- We can use Linux in our embedded projects!!
- Go forth and continue to use and develop Linux
  - I hope you enjoy and learn from the sessions we have these next 3 days



CE Workgroup



**Thanks!**

