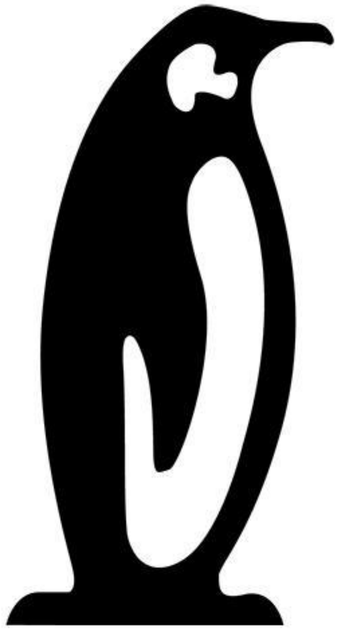
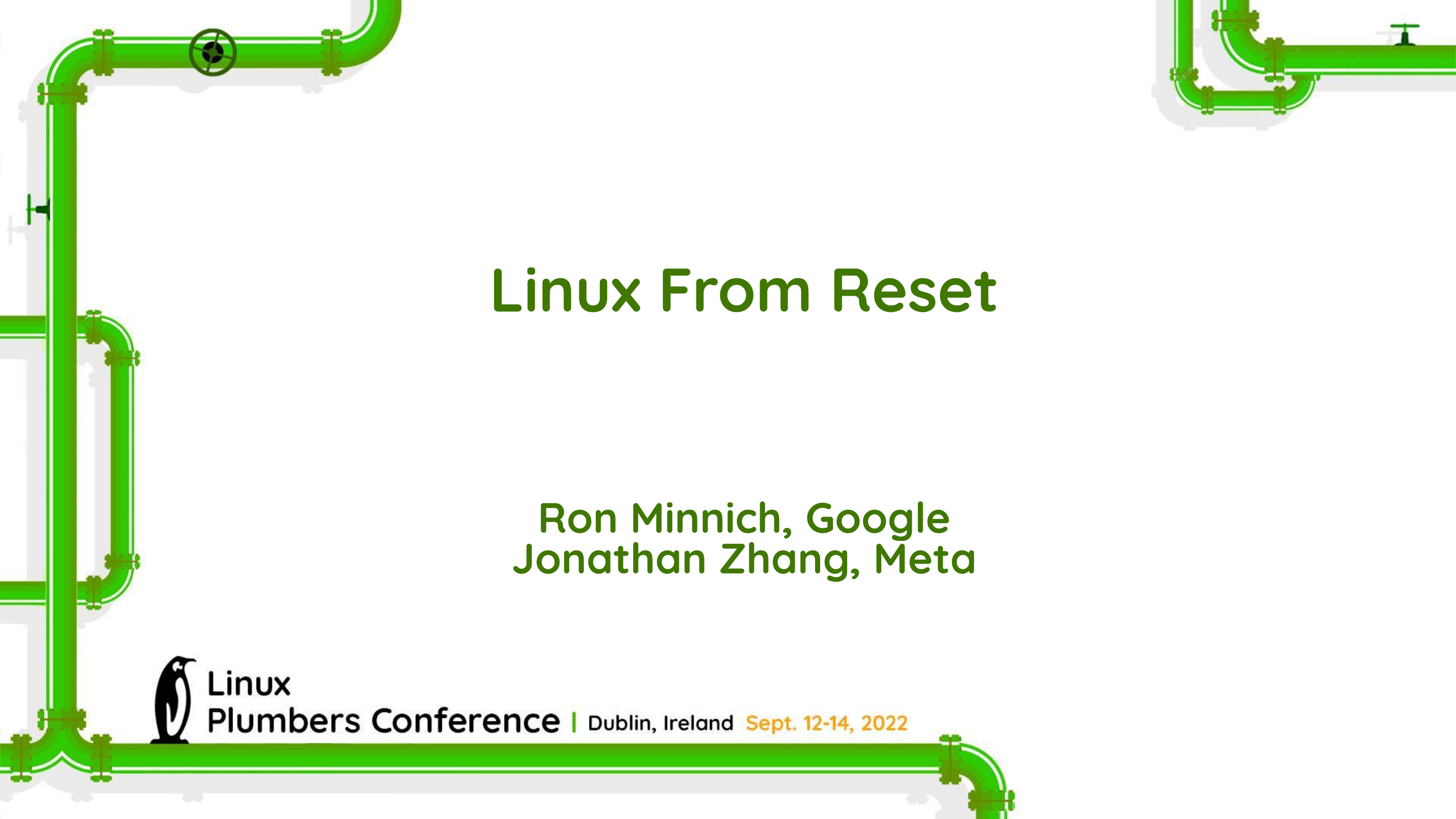


Linux Plumbers Conference

Dublin, Ireland September 12-14, 2022



Linux From Reset

Ron Minnich, Google
Jonathan Zhang, Meta

Linux Plumbers Conference | Dublin, Ireland Sept. 12-14, 2022

Agenda

- Status
- Challenges / Opportunities
- Call to Action

LinuxBoot @Google, ByteDance …

Reset vector → Silicon/Platform Init → Linux Kernel → Go Runtime → Kexec to prodlinux

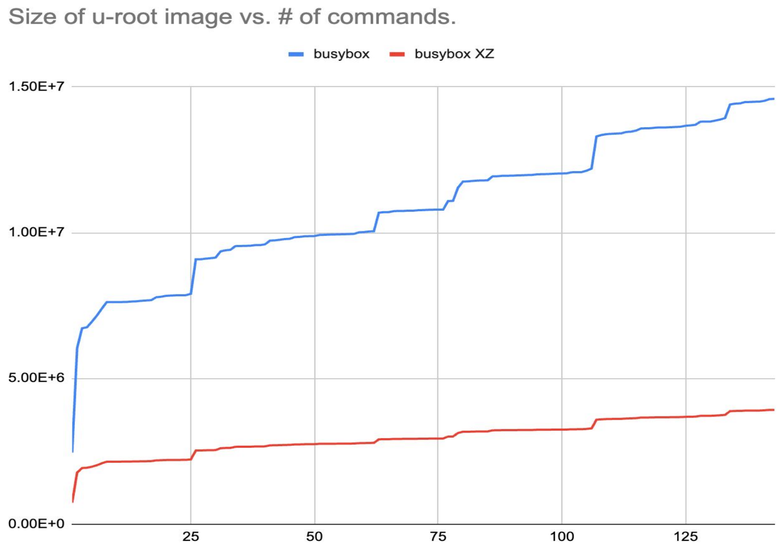
- Linux kernel and runtime in FLASH, replacing much if not all of UEFI
- Deployed to millions of server platform globally, starting from 0 in 2018
- Go runtime uses the Go busybox project for initramfs (including netboot)
  - Gobusybox takes programs, rewrites to packages, compiles to one program
  - Entire rewrite and compile process takes ~10 seconds for 160 programs
  - The initramfs: 160 commands, 18 MiB uncompressed / 5 MiB compressed
- adding diskunlock program: 6 KiB in gobusybox; >10 (can't say) MiB in C

File: ls.go
Package main

Func main()
{
…
}

File: bb/ls.go
Package ls

Func Main()
{
…
}

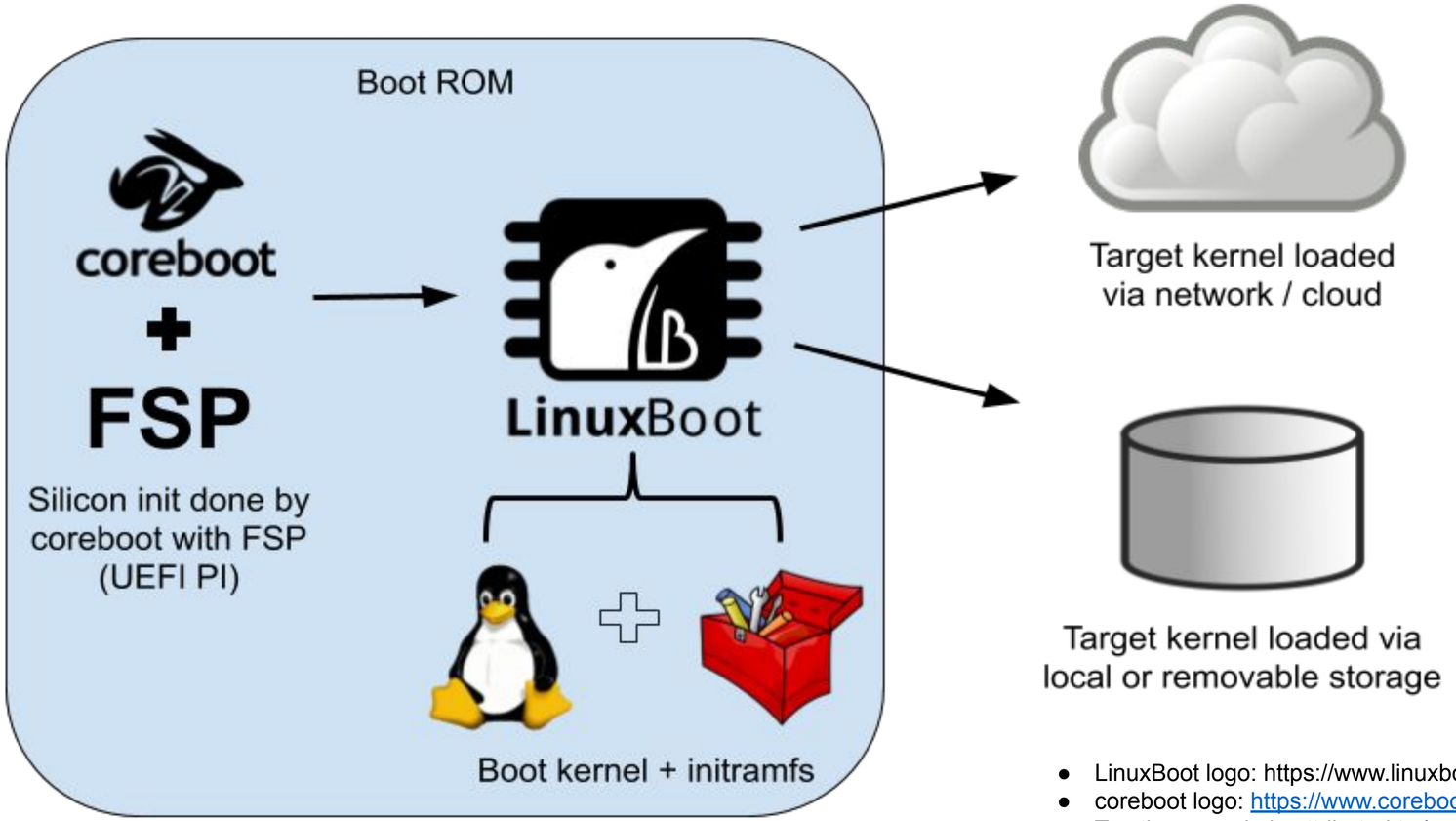


Go Busybox

- Go Busybox lets us use any Go program (https://github.com/u-root/gobusybox)
- Most of the code comes from u-root (https://github.com/u-root/u-root)
- Each u-root PR does build and test across entire project, verifies coverage (74% today), boots x86-64, arm64, arm32 in VMs to test
- Tests, VM tests, and code coverage decrease will block PR

u-root

Cloud Firmware
Note: FSP is a binary blob!

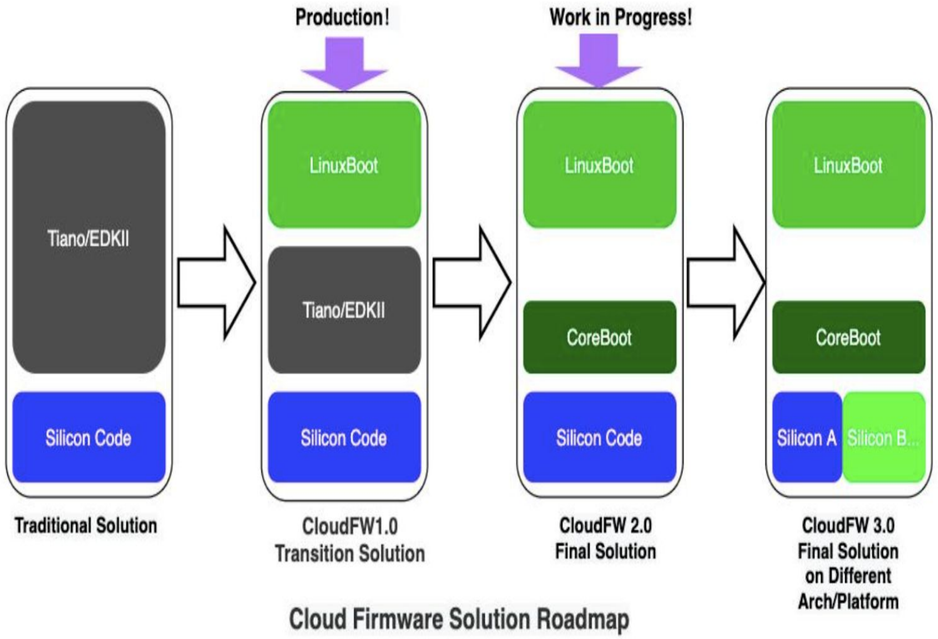


- LinuxBoot logo: https://www.linuxboot.org/page/artwork/
- coreboot logo: https://www.coreboot.org/Logo
- Tux the penguin is attributed to lewing@isc.tamu.edu

coreboot

- Being used in 100 million+ Chromebook devices
  - Supports arm, arm64, ppc64, riscv, x86-32, x86-64
  - 83 chromebook models, 230+ systems, inc. routers, servers.
- Linux kernel inspired community and design philosophy
  - Kconfig build system (ported to coreboot in 2009)
  - One code base support all ISAs, all processors, all servers
    - Obvious as this is, it is not the rule in x86 BIOS world
- code review via Gerrit
  - Full build of 234 mainboards for each CL
- In coverity for over ten years
- Aim to be as slim as possible

Cloud Firmware In Action



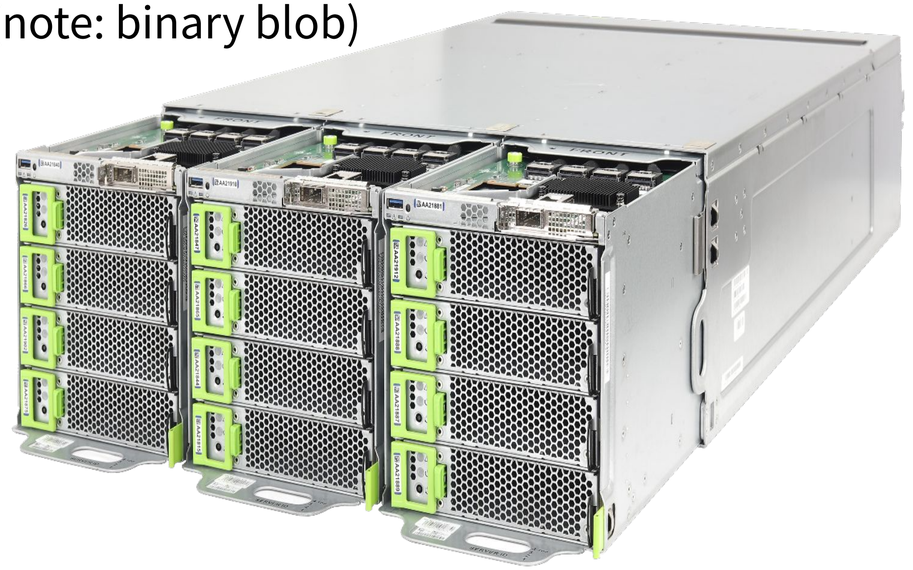
Intel Xeon Scalable Processor	OCP Server	Intel FSP API Mode Status	Status
Sky Lake	Tioga Pass	Proof Of Concept	POC achieved (2020)
Cooper Lake	Delta Lake	Statement Of Work	Pre-production ready achieved, OCP accepted (2021)
Sapphire Rapids	Name to be published	Mainstream Product	Production ready in progress

ByteDance

Meta

Cloud Firmware Ecosystem

- Intel -- foundational software (FSP) on which to build (note: binary blob)
- Hyperscalers
  - Meta
  - ByteDance
  - AWS
- ODMs / OEMs
  - Inspur
  - Quanta
  - SuperMicro
  - WiWynn
  - Lenovo
- Independent Firmware Vendors
  - 9Elements
  - SysPro



- Creating a broad “culture of competence”
- Future ports are easier/faster

Broader Cloud Firmware Support

- AMD Server Platform (goal: blob-free x86; power-on/reset ARM cores remain blobs)
  - Industry coalition formed
  - AMD: POC/POR committed roadmap as a result of internal feasibility studies
  - Oxide: will release full open source Q4 22
  - oreboot port will use Oxide code as “doc” to finish their port
- Arm Server Platform (blob free save for burnt-into-chip “boot block”)
  - Arm SystemReady program supports open source
  - Open source firmware solutions based on LinuxBoot are available
  - Ampere contributed to OCP LinuxBoot for Mt Jade reference system
- RISC-V (blob free on most platforms save for burnt-into-chip “boot block”)
  - Oreboot (Rust) firmware is gaining traction on RISC-V
  - On Allwinner D1-based boards, oreboot provides a full open source stack
  - oreboot boots Linux + Gobusybox initramfs (i.e. LinuxBoot)
  - RISC-V is a great place for rethinking everything
  - note: no real RISC-V server platform – this year. They’re coming.

Agenda

- Status
- Challenges / Opportunities
- Call to Action / Discussion Points

kexec

- Kexec has never been fully capable as a boot loader
  - Continuing problems booting cross-version
- We've had discussions with various distros for years about fixing this
  - We'd like to get it going
- **Goal: Any LTS kernel, starting at (e.g) 4.1x, can boot any newer or older LTS**
- This requires good kexec testing infrastructure
- Fortunately, most of the kexec problems we find are software (toolchain, MSRs, etc) and amenable to cloud approaches
- We have one such approach in progress at Google
- Kexec is load bearing infrastructure:
  - Kernels in FLASH should be able to boot older kernels, e.g. 5.10 should be able to kexec 4.15
  - To keep this requirement manageable, we suggest limiting it to LTS
  - If possible, run non-SMP

MSRs

- Some MSRs, when set, will break older kernels
  - E.g. split lock detection MSR
  - Some MSRs, once set, can not be cleared (e.g. FEATURE)
- Such MSRs might be guarded by CONFIG_LINUXBOOT or a similar mechanism
- Set-once MSRs, if not required, should also be guarded
- Required MSRs: code should be careful to see if they are already set
  - Since some of them have been known to GPF even when the value does not change
  - This is of course a problem for write-only MSRs
    - Perhaps we can convince silicon vendors to stop creating them

Observations

- PCIe
  - Drivers should not require SMP for init!
- Boot time is crucial
  - Use mem= to limit zeroing memory that the next kernel will just zero again
- Security is crucial
  - LinuxBoot can decrypt attestation keys and such
  - Must zero on allocate / zero on free **always**

Observations

- Drivers should be “non-SMP” safe
  - We have seen drivers that can not init unless SMP is enabled
- MSR changes to optimize performance need to be easily reversed, skipped, harmless, or not used at all
  - Split cache MSR will panic older kernels when they kexec
- A known good toolchain should be specified in kernel configs
  - I.e. COMPILER_PREFIX should *never* be empty
  - Coreboot has always specified the correct toolchain for every release; Linux probably should too
- Firmware Tables must be “stable” – kexec does this right for e820/ACPI

Agenda

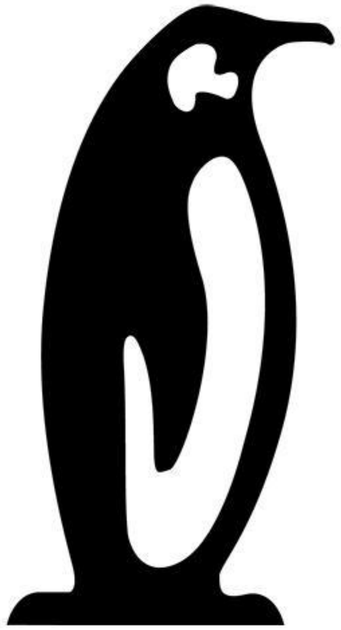
- Status
- Challenges / Opportunities
- Call to Action / Discussion Points

Call to Action

- Use Open Source host Firmware to:
  - debug kernel/OS issues related to host firmware
  - verify that Linux can boot with open source firmware
  - Fix host firmware issues
  - Improve end to end solution
  - Provide reference behavior
- Engage Open Source Firmware community:
  - Open Source Firmware Foundation
  - OCP Open System Firmware Project
    - What's the difference? OCP allows silicon vendor blobs
  - coreboot/LinuxBoot mailing lists and slack channel(s)

Discussion Points

- How to Collaborate
  - Host firmware community
  - Kernel community
  - OS Vendors



Linux Plumbers Conference

Dublin, Ireland September 12-14, 2022