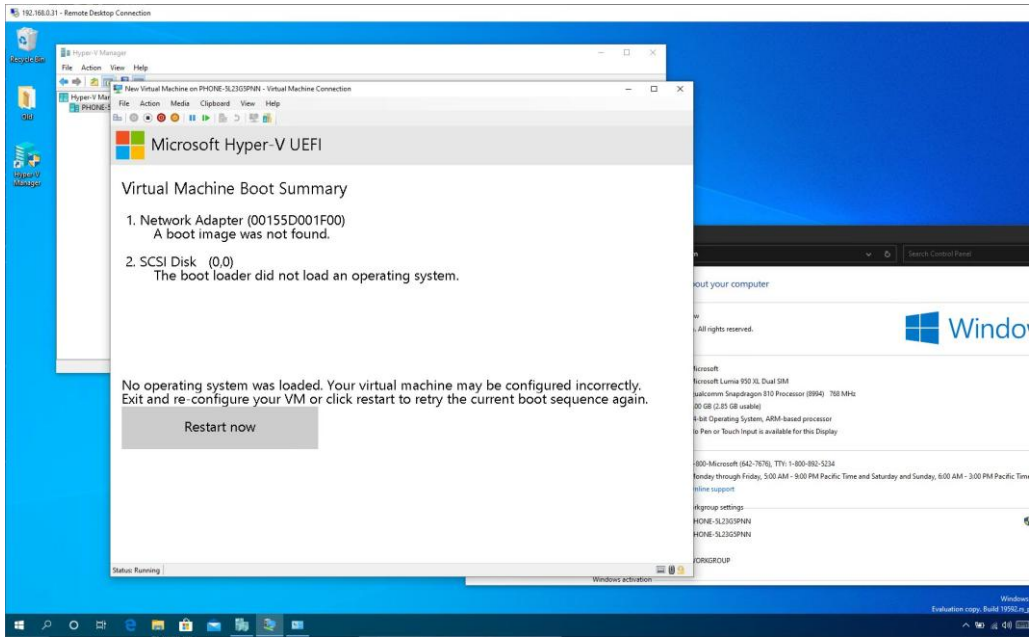
“我的 Windows Phone 果然有问题”

Modern AArch64 SoC
UEFI + ACPI port from A to Z



Bingxing Wang (@imbushuo), 5/16/2020

imbushuo.net

Post your questions to *hi(at)imbushuo.net*

Also the team: Gustave Monce (@gus33000), ADeltaX, José Manuel Nieto (@SuperJMN), Googulator, CornyjK, Dj Art, Simone Franco, Tiger Wang and woachk

Table of Contents

Non-goal

- OS-specific driver bring-up and implementation

About me

- Final-year college student
- Site Reliability Engineer Intern @ LinkedIn
  - Technically it is Microsoft
- Firmware and hardware hacks for fun
  - I can’t play games well
- “外国黑客” @ ITHome

Modern mobile SoC boot flow

- Modern mobile SoC is complicated
  - Rich features and connectivity
  - Performance improvements every year
  - When in doubt, add one more general-purpose CPU
  - There are more than 12 CPUs in Snapdragon 810
    - AP(Cortex A53 & A57), Modem and GNSS(QDSP6), Audio (Xtensa), Video(Cortex-A), GPS(Cortex-A), Power (Cortex-M)
- Boot the SoC
  - Subsystems have dependencies
  - Can’t boot them all at once
  - Static Root of Trust & trust chain for boot security

UEFI + ACPI on AArch64

Common ARM bootstrap situation

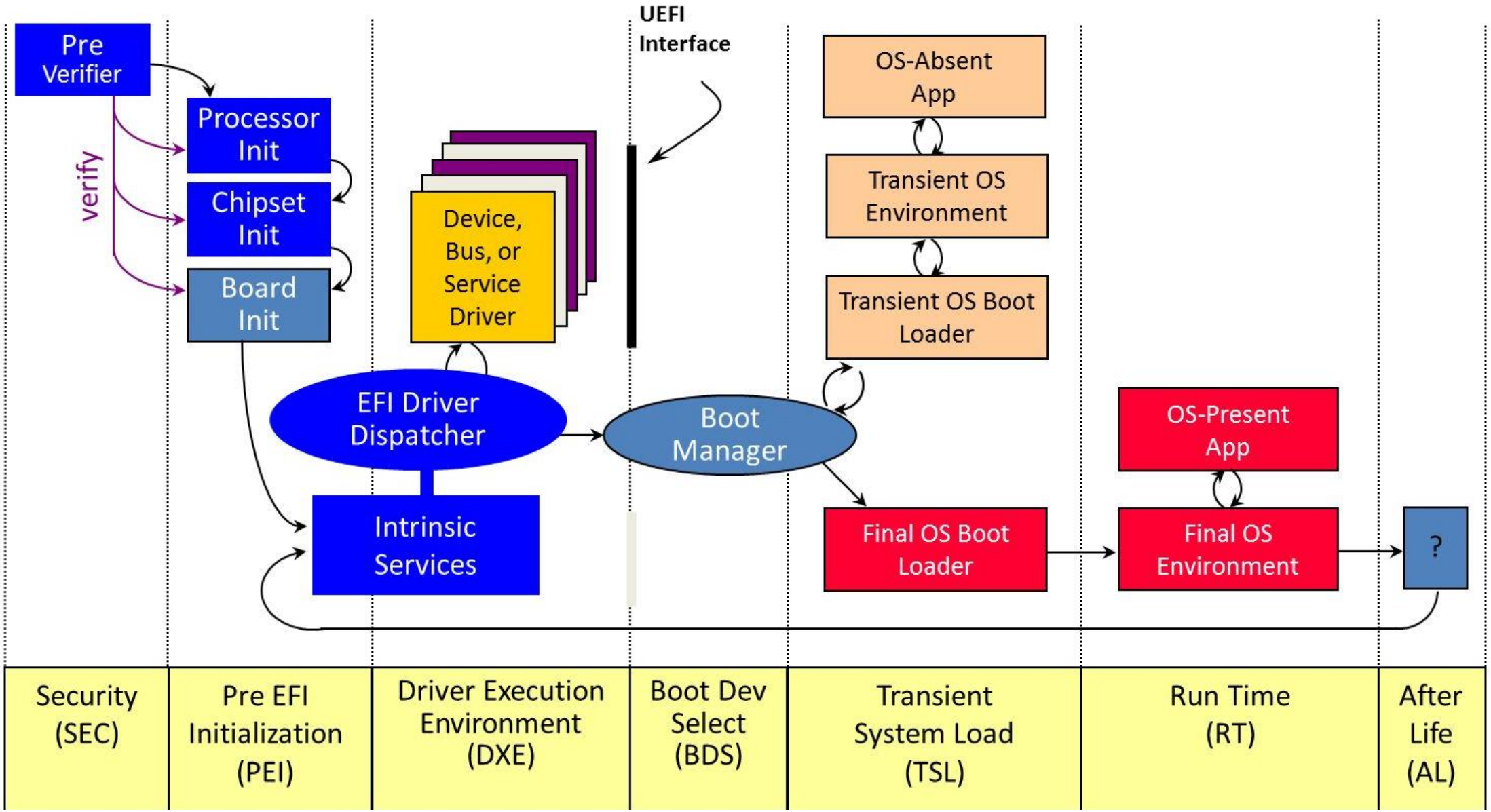
- Platform-specific images
  - No generic image for all devices
- Board init file or device tree for topology
  - Because “embedded”
  - No flexibility in hardware configuration
- U-Boot is a popular choice
  - For very long time and it continues
  - Also implemented UEFI recently: https://www.suse.com/media/article/UEFI_on_Top_of_U-Boot.pdf
  - Enough to boot GRUB, not targets Windows yet

UEFI and ACPI

- Existing (yet broken) industry standard on PC for years
- A defined set of interfaces and protocols for OS interaction
  - Like BIOS interrupt on PC/AT
- Commercially introduced to ARM platform with Windows RT (ACPI 5.0)
  - Linux implemented UEFI support for both AArch32 and AArch64
  - Either Device Tree or ACPI for hardware description
  - But Linux ARM ACPI is AArch64 exclusive
- Enhanced with AArch64 development recently
- Try UEFI on Raspberry Pi 3 and 4
  - https://rpi4-uefi.dev/
  - https://connect.linaro.org/resources/ltd20/ltd20-207/

UEFI vs. U-Boot

- U-Boot optimizes for size
  - U-Boot links built-in things together for speed and size
  - U-Boot features and environments can be customized
  - Single file for each phase, can be stripped
  - Versatile payloads
- UEFI targets generalization
  - LZMA-compressed, firmware volume
  - UEFI drivers are PE/COFF executables
  - UEFI Drivers usually run fine on another implementation
  - Most UEFI implementations have a baseline for protocols – a lot
  - EFI applications


Pre Verifier
verify
Processor Init
Chipset Init
Board Init
Device, Bus, or Service Driver
UEFI Interface
OS-Absent App
Transient OS Environment
Transient OS Boot Loader
EFI Driver Dispatcher
Boot Manager
Intrinsic Services
Final OS Boot Loader
OS-Present App
Final OS Environment
?
Security (SEC)
Pre EFI Initialization (PEI)
Driver Execution Environment (DXE)
Boot Dev Select (BDS)
Transient System Load (TSL)
Run Time (RT)
After Life (AL)

ACPI on ARM

- Hardware-reduced ACPI: just a bunch of ACPI tables
- Fixed ACPI tables and DSDT
  - MADT – Generic Interrupt Controller
  - GTDT – Architectural Timer
  - CSRT – Vendor IP cores such as DMA controller
  - IORT – IOMMU redirection table
  - PPTT – Processor topology
  - MCFG – PCIe MMIO
  - FACS – Firmware control
  - FADT – Fixed ACPI description
  - …

ACPI static tables

```
[02Ch 0044   1]                Subtable Type : 0B [Generic Interrupt Controller]
[02Dh 0045   1]                       Length : 50
[02Eh 0046   2]                     Reserved : 0000
[030h 0048   4]         CPU Interface Number : 00000000
[034h 0052   4]                Processor UID : 00000000
[038h 0056   4]        Flags (decoded below) : 00000001
                             Processor Enabled : 1
          Performance Interrupt Trigger Mode : 0
          Virtual GIC Interrupt Trigger Mode : 0
[03Ch 0060   4]     Parking Protocol Version : 00000001
[040h 0064   4]        Performance Interrupt : 00000017
[044h 0068   8]                Parked Address : 0000000000301000
[04Ch 0076   8]                  Base Address : 00000000F9002000
[054h 0084   8]     Virtual GIC Base Address : 00000000F9004000
[05Ch 0092   8]  Hypervisor GIC Base Address : 00000000F9001000
[064h 0100   4]        Virtual GIC Interrupt : 00000019
[068h 0104   8]   Redistributor Base Address : 0000000000000000
[070h 0112   8]                     ARM MPIDR : 0000000000000000
[078h 0120   1]             Efficiency Class : 00
[079h 0121   3]                     Reserved : 000000
```

```
intc: interrupt-controller@f9000000 {
        compatible = "qcom,msm-qgic2";
        interrupt-controller;
        #interrupt-cells = <3>;
        reg = <0xf9000000 0x1000>,
                <0xf9002000 0x1000>;
};
```

DSDT table

- DSDT contains both topology information and optional program logics
- Bytecode-based DSL
  - OS runs it
  - Turing complete
  - Memory I/O with bytecode instructions
  - Capable to handle multiple OS scenarios
- Dynamic updates with SSDT tables
  - Device Tree is usually a static table (with optional overlays)

```
Device (SDC1)
{
    Name (_DEP, Package (One)  // _DEP: Dependencies
    {
        \_SB.PEP0
    })
    Name (_HID, "QCOM24BF")  // _HID: Hardware ID
    Name (_CID, "ACPIQCOM24BF")  // _CID: Compatible ID
    Name (_UID, Zero)  // _UID: Unique ID
    Name (_CCA, Zero)  // _CCA: Cache Coherency Attribute
    Method (_CRS, 0, NotSerialized)  // _CRS: Current Resourc
    {
        Name (RBUF, ResourceTemplate ()
        {
            Memory32Fixed (ReadWrite,
                0xF9824900,         // Address Base
                0x00000200,         // Address Length
                )
            Interrupt (ResourceConsumer, Level, ActiveHigh, E
            {
                0x0000009B,
            }
        })
        Return (RBUF) /* \_SB_.SDC1._CRS.RBUF */
    }

    Device (EMMC)
    {…
    }

    Method (_DIS, 0, NotSerialized)  // _DIS: Disable Device
    {
    }

    Method (_STA, 0, NotSerialized)  // _STA: Status
    {…
    }
}
```

A sample DSDT device object

- Dependency entries
- ID entries
- Cache coherency attribute for DMA
- Resource settings – MMIO, interrupt, GPIO, …
- ACPI methods – status, …
- Sub-devices

Lumia 950 (XL)

- Snapdragon 808 / 810 SoC
  - Cortex A53 + A57, AArch64
- Windows Phone OS never goes AArch64
  - EL1 runs in AArch32 mode
  - But EL2 hypervisor and EL3 are AArch64

```
$ file HYP.img
HYP.img: ELF 64-bit LSB executable, ARM aarch64,
version 1 (SYSV), statically linked, stripped
```

Finding Qualcomm proprietary docs…

Your NDA signing is a hall of shame

Dig into vendor blobs

UEFITool NE alpha 47 - UEFI.img

File Action View Help

Structure

Name	Act	Type	Subtype	Text
UEFI image		Image	UEFI	
Padding		Padding	Non-empty	
EfiFirmwareFileSystem2G…		Volume	FFSv2	
8AF09F13-44C5-96EC-143…		File	SEC core	
DDE58710-41CD-4306-DBF…		File	Freeform	uefiplat.cfg
9E21FD93-9C72-4C15-8C4…		File	Volume image	
LzmaCustomDecompressG…		Section	GUID defined	
Raw section		Section	Raw	
Volume image section		Section	Volume image	
EfiFirmwareFileSyste…		Volume	FFSv2	
AprioriDxe		File	Freeform	DXE apriori file
DxeCore		File	DXE core	DxeCore
ArmCpuDxe		File	DXE driver	ArmCpuDxe
RuntimeDxe		File	DXE driver	RuntimeDxe
5E0EAE60-EAED-4D75-…		File	DXE driver	SecurityDxe
WatchDogTimerDxe		File	DXE driver	WatchdogTimer
55CE7A0C-5598-4B1F-…		File	DXE driver	CapsuleRuntimeDxe
2B0ECDCE-01AE-446E-…		File	DXE driver	VariableDxe
65B852DF-355E-4946-…		File	DXE driver	DppDxe
37795BA0-E1CF-4ED5-…		File	DXE driver	EmbeddedMonotonicCounter

Parser FIT BootGuard Search Builder

```
parseDepexSectionBody: unknown opcode
parseDepexSectionBody: unknown opcode
parseDepexSectionBody: unknown opcode
parse: not a single Volume Top File is found, the image may be corrupted
```

Information

```
ZeroVector:
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
Signature: _FVH
FileSystem GUID:
8C8CE578-8A3D-4F1C-9935-896185C3
```

```
[Config]
Version = 3
MaxMemoryRegions = 64

[MemoryMap]
#                                                    EFI_RESOURCE_ EFI_RESOURCE_ATTRIBUTE_        ARM_REGION_ATTRIBUTE_
#MemBase,   MemSize,   MemLabel(32 Char.), BuildHob, ResourceType, ResourceAttribute, MemoryType, CacheAttributes

#------------ DDR Regions ------
0x00010000, 0x00014000, "DBI Dump",          NoHob,  MMAP_IO, INITIALIZED,  Conv,   NS_DEVICE
0x00024000, 0x00002000, "DDR Health Mon",    NoHob,  MMAP_IO, INITIALIZED,  Reserv, NS_DEVICE
0x00100000, 0x00100000, "HLOS 0",            AddMem, SYS_MEM, SYS_MEM_CAP,  Conv,   WRITE_BACK
0x00200000, 0x00100000, "UEFI FD",           AddMem, SYS_MEM, SYS_MEM_CAP,  BsCode, WRITE_BACK
0x00300000, 0x00080000, "MPPark Code",       AddMem, MEM_RES, UNCACHEABLE,  RtCode, UNCACHED_UNBUFFERED
0x00380000, 0x00001000, "FBPT Payload",      AddMem, SYS_MEM, SYS_MEM_CAP,  RtData, UNCACHED_UNBUFFERED
0x00381000, 0x00004000, "DBG2",              AddMem, SYS_MEM, SYS_MEM_CAP,  LdData, UNCACHED_UNBUFFERED
0x00385000, 0x00001000, "Capsule Header",    AddMem, SYS_MEM, SYS_MEM_CAP,  RtData, UNCACHED_UNBUFFERED
0x00386000, 0x00003000, "TPM Control Area",  AddMem, SYS_MEM, SYS_MEM_CAP,  RtData, UNCACHED_UNBUFFERED
0x00389000, 0x00001000, "UEFI Info Block",   AddMem, SYS_MEM, SYS_MEM_CAP,  RtData, UNCACHED_UNBUFFERED
0x0038A000, 0x00003000, "Reset Data",        AddMem, SYS_MEM, SYS_MEM_CAP,  RtData, UNCACHED_UNBUFFERED
0x0038D000, 0x00073000, "Reser. Uncached0",  AddMem, SYS_MEM, SYS_MEM_CAP,  BsData, UNCACHED_UNBUFFERED
0x00400000, 0x00800000, "Display Reserved",  AddMem, MEM_RES, WRITE_THROUGH, MaxMem, WRITE_THROUGH
0x00C00000, 0x00040000, "UEFI Stack",        AddMem, SYS_MEM, SYS_MEM_CAP,  BsData, WRITE_BACK
0x00C40000, 0x00010000, "CPU Vectors",       AddMem, SYS_MEM, SYS_MEM_CAP,  BsCode, WRITE_BACK
0x00C50000, 0x000B0000, "Reser. Cached 0",   AddMem, SYS_MEM, SYS_MEM_CAP,  BsData, WRITE_BACK
0x00D00000, 0x03300000, "HLOS 1",            AddMem, SYS_MEM, SYS_MEM_CAP,  BsData, WRITE_BACK
0x04000000, 0x02500000, "HLOS 2",            AddMem, SYS_MEM, SYS_MEM_CAP,  Conv,   WRITE_BACK
0x06500000, 0x00500000, "TZ Apps",           AddMem, SYS_MEM, SYS_MEM_CAP,  Reserv, NS_DEVICE
0x06A00000, 0x00200000, "SMEM",              AddMem, MEM_RES, UNCACHEABLE,  Reserv, UNCACHED_UNBUFFERED
0x06C00000, 0x00100000, "Hypervisor",        AddMem, SYS_MEM, SYS_MEM_CAP,  Reserv, NS_DEVICE
0x06D00000, 0x00200000, "TZ",                AddMem, SYS_MEM, SYS_MEM_CAP,  Reserv, NS_DEVICE
0x06F00000, 0x00180000, "MPSS_EFS / SBL",    AddMem, SYS_MEM, SYS_MEM_CAP,  Reserv, NS_DEVICE
0x07080000, 0x00020000, "ADSP_EFS",          AddMem, SYS_MEM, SYS_MEM_CAP,  Reserv, NS_DEVICE
```

https://github.com/LongSoft/UEFITool

Open source efforts

Linaro

index : working/qualcomm/lk.git

[no description]

mast

summary | refs | log | tree | commit | diff

Branch	Commit message
release/LA.AF.1.1-02810-8064+rescue	Fastboot: Moving forcefully board into fastboot
release/LA.BR.1.1.2-02210-8x16.0	msm8916: lk: Android SD Card boot
release/LA.BR.1.1.2-02210-8x16.0+rescue	aboot: force boot into fastboot
release/LA.BR.1.1.2-02210-8x16.0+sdboot	aboot: set local-mac-address in DTS
release/LA.BR.1.2.4-00310-8x16.0	aboot: set local-mac-address in DTS
release/LA.BR.1.2.7-03810-8x16.0	platform/msm8916: Allow bigger kernels
release/LA.BR.1.2.7-03810-8x16.0+rescue	aboot: force boot into fastboot
release/LA.BR.1.2.7-03810-8x16.0+sdboot	dev-tree: Support non-skales DTB if only one appended
release/LA.HB.1.3.2-19600-8x96.0	platform/msm8996: Increase memory area for load kernel
release/LA.HB.1.3.2-19600-8x96.0+rescue	assert: ensure DEBUG_LEVEL is set before it is used

[...]

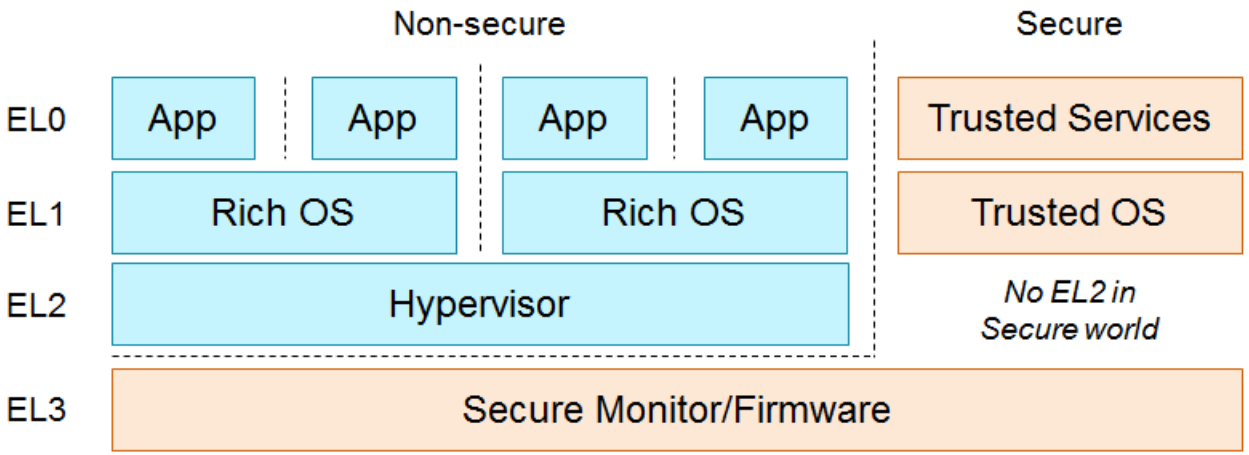
Tag	Download
dragonboard410c-LA.BR.1.2.7-03810-8x16.0-linaro3	lk-dragonboard410c-LA.BR.1.2.7-03810-8x16.0-linaro3.tar.gz
dragonboard410c-LA.BR.1.2.7-03810-8x16.0-linaro2	lk-dragonboard410c-LA.BR.1.2.7-03810-8x16.0-linaro2.tar.gz
dragonboard410c-LA.BR.1.2.7-03810-8x16.0-linaro1	lk-dragonboard410c-LA.BR.1.2.7-03810-8x16.0-linaro1.tar.gz
debian-qcom-dragonboard410c-LA.BR.1.2.4-00310-8x16.0-linaro2	lk-debian-qcom-dragonboard410c-LA.BR.1.2.4-00310-8x16.0-linaro2.tar.gz
debian-qcom-dragonboard410c-LA.BR.1.2.4-00310-8x16.0-linaro1	lk-debian-qcom-dragonboard410c-LA.BR.1.2.4-00310-8x16.0-linaro1.tar.gz
ubuntu-qcom-dragonboard410c-LA.BR.1.2.4-00310-8x16.0-linaro1	lk-ubuntu-qcom-dragonboard410c-LA.BR.1.2.4-00310-8x16.0-linaro1.tar.gz

Age	Commit message
2015-06-02	Merge "arch: arm: Clean and invalidate by set/way instead of invalidate set/way" HEAD master
2015-06-02	Merge "target: msm8996: support dsc panel selection through fastboot"
2015-06-02	Merge "app: aboot: Fix wrong message of devinfo partition"
2015-06-01	Merge "platform: msm8994: Make scratch memory MMU cache setting to WRITE_BACK...
2015-06-01	Merge "target: msm8909: update the DSI PLL enable sequence for 8909"
2015-06-01	Merge "platform: msm8952: Update the usb frequency"
2015-06-01	arch: arm: Clean and invalidate by set/way instead of invalidate set/way
2015-06-01	platform: msm8994: Make scratch memory MMU cache setting to WRITE_BACK_ALLOCATE
2015-05-31	target: msm8952: Add LDO1 for splash screen support
2015-05-31	platform: msm_shared: Add macro for LDO1

[...]

https://git.linaro.org/landing-teams/working/qualcomm/lk.git

AArch64 Exception Levels

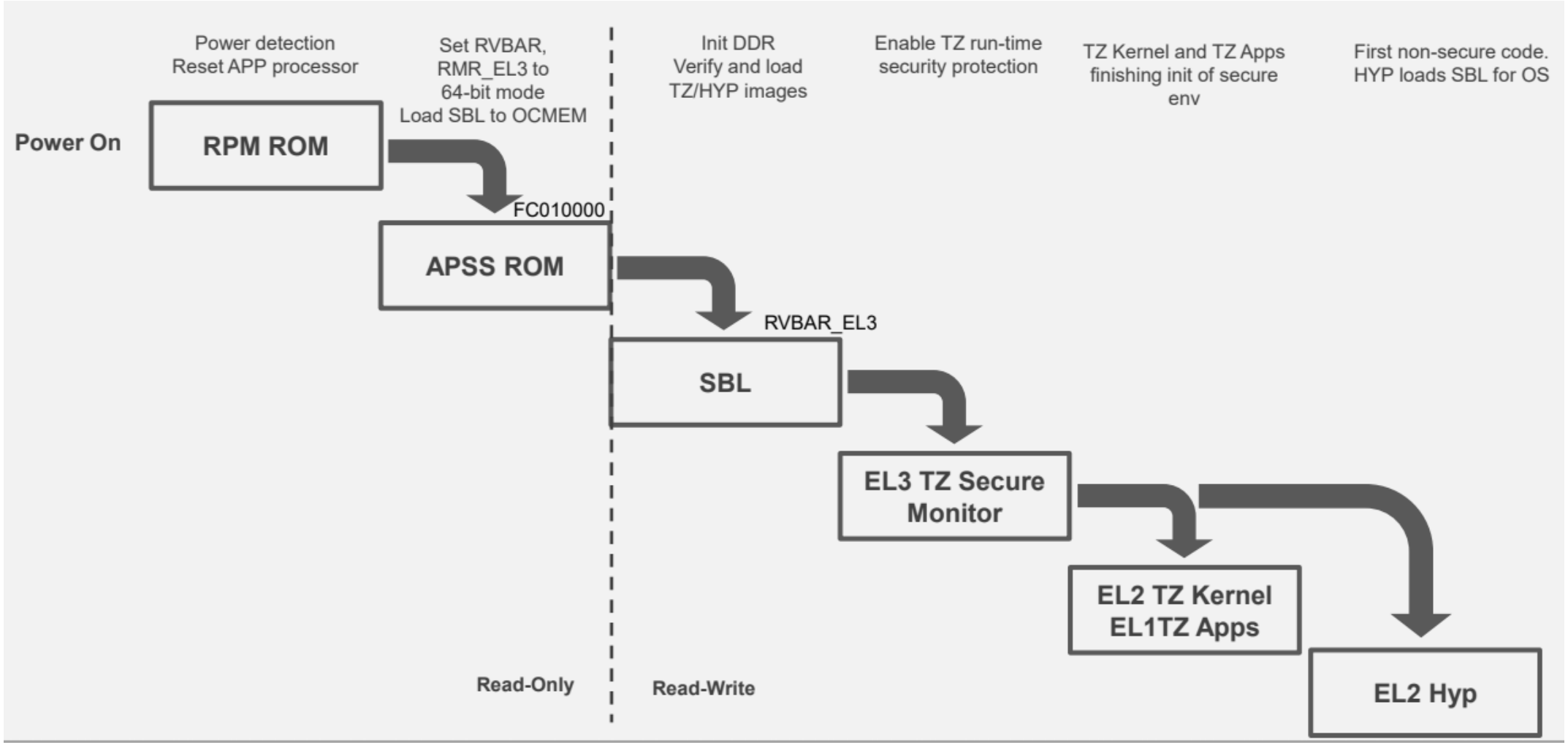


** Recently Secure EL2 (Secure partition manager) is introduced*

Snapdragon 810 Boot Flow

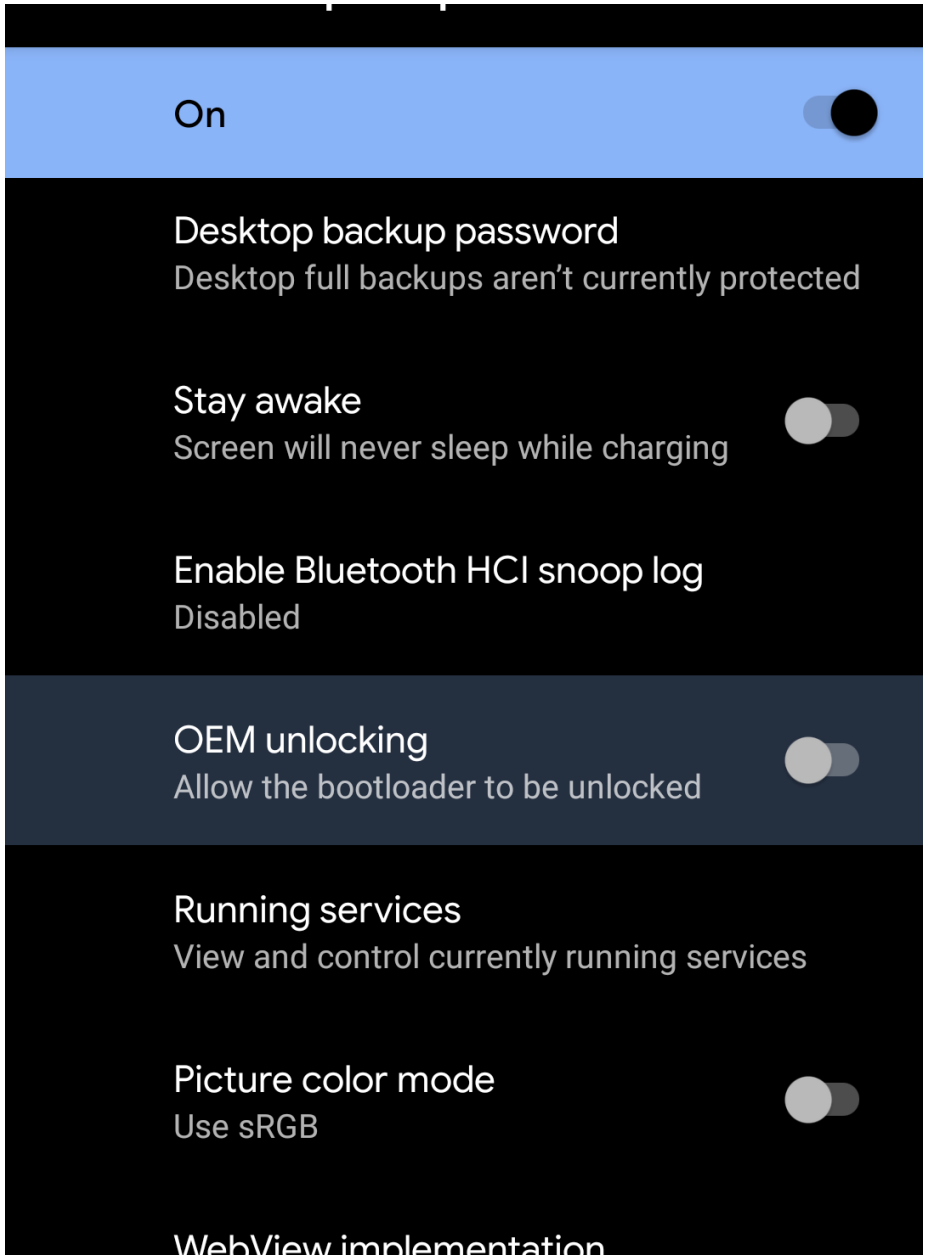
- Boot from eMMC or UFS
  - GPT-based partitioning
  - Partition layout is mostly identical
- Vendor blobs partition and system partition
  - SBL, Modem, DSP, Hypervisor, TrustZone Monitor, UEFI
  - EFI system partition, Windows OS partition
- RPM boots from hardwired PBL, PBL loads SBL
- Final stage bootloader loads OS
  - OS boots subsystems (Audio, Modem, GNSS, GPU, DSP)
  - TrustZone RPC calls

Snapdragon 810 Boot Flow (cont.)



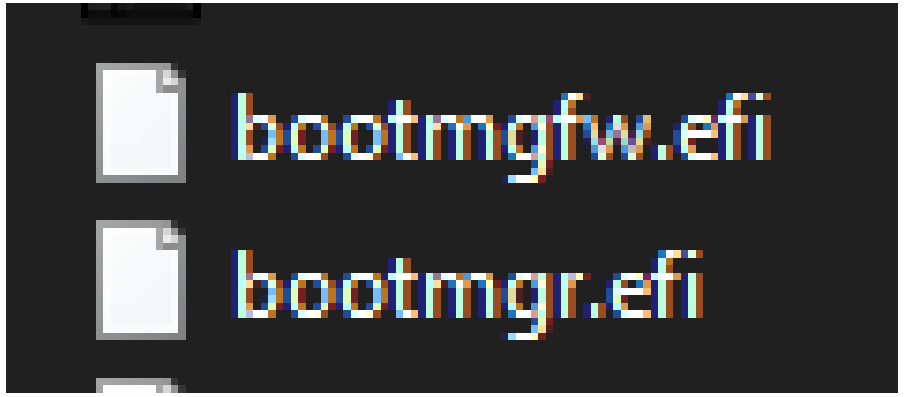
Getting code execution

- “OEM unlock” on Android devices
  - Allows arbitrary code execution in EL1
  - Does not allow modifying bootloader
  - Windows Phones are locked down in production



Getting code execution (cont.)

- Nokia (MMO) exploit allows arbitrary flash I/O
- Qualcomm UEFI variable services exploit
  - UEFI spec uses variable to report Secure Boot status
  - Secure Boot status variable is expected to be volatile
  - Let's put *SecureBoot = 0* in the variable storage…
  - Firmware reports Secure Boot off to Windows Boot Manager (!)
  - Can't run EFI application payloads due to additional checking
- https://github.com/ReneLergner/WPinternals
- Seize control from Windows Boot Manager


bootmgfw.efi
bootmgr.efi


CONFUSED FIRMWARE
BOOTMGR.EFI
IS THAT A UEFI APPLICATION
imgflip.com

/SUBSYSTEM (Specify Subsystem)

11/04/2016 • 2 minutes to read •

Copy

```
/SUBSYSTEM:{BOOT_APPLICATION|CONSOLE|EFI_APPLICATION|
            EFI_BOOT_SERVICE_DRIVER|EFI_ROM|EFI_RUNTIME_DRIVER|NATIVE|
            POSIX|WINDOWS)
            [,major[.minor]]
```

Arguments

BOOT_APPLICATION

An application that runs in the Windows boot environment. For more information about boot applications, see About BCD.

CONSOLE

Win32 character-mode application. The operating system provides a console for console applications. If `main` or `wmain` is defined for native code, `int main(array<String ^> ^)` is defined for managed code, or you build the application completely by using `/clr:safe`, CONSOLE is the default.

EFI_APPLICATION

EFI_BOOT_SERVICE_DRIVER

EFI_ROM

EFI_RUNTIME_DRIVER

Is this page helpful?

Yes No

In this article

Arguments

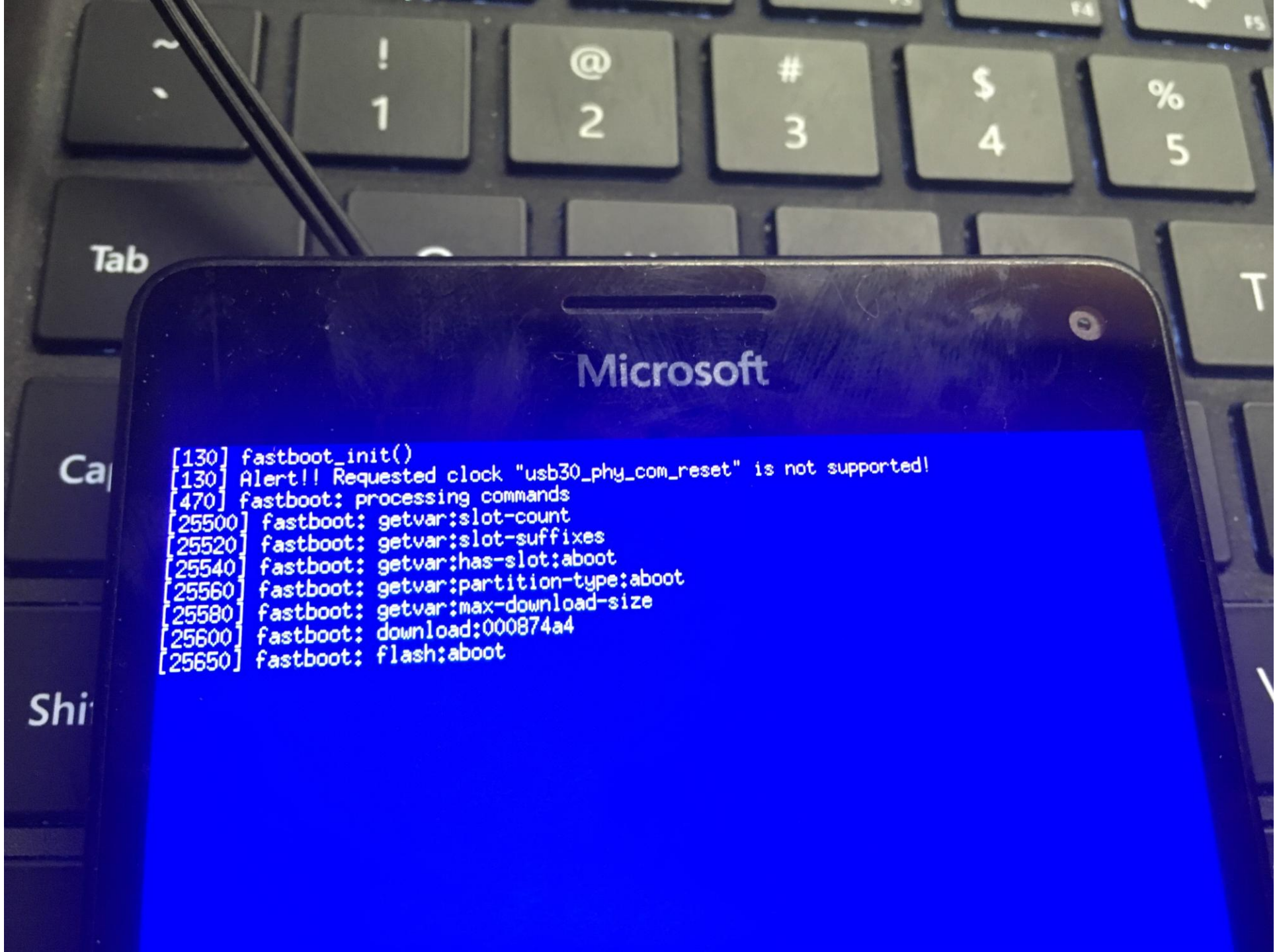
Remarks

See also

```
// We are currently in the APPLICATION context.
// We need to switch to the Firmware context FIRST otherwise things
unsigned int InterruptState = FirmwareDescriptor->InterruptState;

// Disable IRQ
DisableInterrupt();

// First up, switch MM state
unsigned long Value = FirmwareDescriptor->MmState.HardwarePageDirect
        FirmwareDescriptor->MmState.TTB_Config;

ArmMoveToProcessor(Value, CP15_TTBR0);
ArmInstructionSynchronizationBarrier();

ArmMoveToProcessor(0, CP15_TLBIALL);
ArmInvalidateBTAC();
ArmDataSynchronizationBarrier();
ArmInstructionSynchronizationBarrier();

// Next up, set the exception state
ArmMoveToProcessor(FirmwareDescriptor->ExceptionState.IdSvcRW, CP15_
ArmDataSynchronizationBarrier();

ArmMoveToProcessor(FirmwareDescriptor->ExceptionState.Control, CP15_
ArmInvalidateBTAC();
ArmDataSynchronizationBarrier();
ArmInstructionSynchronizationBarrier();

ArmMoveToProcessor(FirmwareDescriptor->ExceptionState.Vbar, CP15_VBA
ArmInstructionSynchronizationBarrier();

// Restore IRQ if necessary
if (InterruptState) ArmEnableInterrupt();
```

Getting code execution (cont.)

- Windows Boot Manager initializes its own environment
  - MMU, exception vector, …
  - A struct used to describe the firmware context and app context
  - Switch context to access UEFI services
- Secret Secure Monitor call to enter AArch64 EL1
  - Secret SMC call for switching from AArch32 to AArch64
  - Specifies entry point address and switch
- https://github.com/imbushuo/boot-shim


Microsoft
[130] fastboot_init()
[130] Alert!! Requested clock "usb30_phy_com_reset" is not supported!
[470] fastboot: processing commands
[25500] fastboot: getvar:slot-count
[25520] fastboot: getvar:slot-suffixes
[25540] fastboot: getvar:has-slot:aboot
[25560] fastboot: getvar:partition-type:aboot
[25580] fastboot: getvar:max-download-size
[25600] fastboot: download:000874a4
[25650] fastboot: flash:aboot


Microsoft
[90] Recovery command: 32 .
[90] Unable to locate /bootselect partition
[90] Run ELF64 boot routine
[100] Verifier: Payload has valid ELF magic.
[100] Verifier: ELF reports valid architecture.
[100] Verifier: ELF reports valid type.
[110] Verifier: 1 program header entries found.
[110] FD entry point = 0x200000, partition offset = 0x10000, size = 0x120000
[120] UEFI FD loaded into memory
[130] UEFI FD ready.
[130] Channel alloc freed
[140] Jumping to kernel via monitor
Microsoft

Port TianoCore to new platform

- Toolchains
  - Linaro GCC or any recent AArch64 GCC
  - Clang/LLVM should work
- Platform docs and firmware layout
  - Set required package properties in .DSC and .FDF file
- SEC/PEI
  - ARM Trusted Firmware
  - SoC-specific loaders
- DXE
  - Port drivers

TianoCore/EDK2 directory structure

- Feature/Silicon support modules are categorized as “packages” – ArmPkg, MdePkg, MdeModulePkg, …
- Top level packages contain applications, libraries and blob resources
- Package manifest (.DSC and .DEC) provides metadata information
  - GUID/Protocol tokens
  - C Header directories
  - Config key-value pairs
  - Customized tools
  - …
- Firmware manifest (.FDF) describes FD layout
  - The firmware package
  - Executable

.azurepipelines	.azurepipelines: Add RISC-V architecture on RISC-V EDK2 CI.	7 days ago
.mergify	.mergify: Add Mergify YML pull request rules configuration file	6 months ago
.pytool	.pytool/CISettings: Remove Windows only scope for host based unit tests	6 days ago
ArmPkg	ArmPkg/MmCommunicationDxe: expose MM Communicate 2 protocol	yesterday
ArmPlatformPkg	ArmPlatformPkg: remove PL180 SD controller driver	9 days ago
ArmVirtPkg	ArmVirtPkg: control PXEv4 / PXEv6 boot support fro	
BaseTools	BaseTools: Fix parse PCD GUID expression issue	
Conf	BaseTools:Delete FrameworkDatabase from BaseTo	
CryptoPkg	CryptoPkg/Pkcs7: Extend support for other OID type	
DynamicTablesPkg	DynamicTablesPkg: SRAT: Fix uninitialized memory u	
EmbeddedPkg	EmbeddedPkg/EmbeddedPkg.dsc: remove some sta	
EmulatorPkg	EmulatorPkg: Add Platform CI and configuration for	
FatPkg	FatPkg: Add RISC-V architecture for EDK2 CI.	
FmpDevicePkg	FmpDevicePkg/FmpDxe: Fix uninitialized pointer de	
IntelFsp2Pkg	IntelFsp2Pkg/SplitFspBin.py: Coverity scan flags issu	
IntelFsp2WrapperPkg	IntelFsp2WrapperPkg: Fix various typos	
MdeModulePkg	MdeModulePkg/VariableSmmRuntimeDxe: switch to	
MdePkg	MdePkg: introduce MM communicate 2 protocol	
NetworkPkg	NetworkPkg: Add RISCV64 architecture	

Branch: master edk2 / ArmPkg /

Ard Biesheuvel and mergify ArmPkg/MmCom

..

Drivers	ArmPkg/
Filesystem/SemihostFs	ArmPkg:
Include	ArmPkg/
Library	ArmPkg/
ArmPkg.dec	ArmPkg/
ArmPkg.dsc	ArmPkg/

Directory structure (Cont.)

- In this implementation:
  - ACPI table sources & blobs
  - Peripheral drivers
  - Support libraries
  - PEI initialization code
  - CI scripts
  - ELF wrapper for FD file

```
ENTRY(_start);

SECTIONS
{
    _start = 0x00200000;
    . = 0x00200000;
    .data : {
        *(.data)
    }
}
```

gus33000 ACPI: [8992] set vendor to QCOMEDK2 to apply Windows erratum (#63)

AcpiTables	ACPI: [8992] set vendor to QCOMEDK2 to apply Windows erratun
Application	BootApp: clear prompt before booting apps
Driver	RMI4: touch log is info only [skip ci]
GPLDriver	Lattice: UC120 FPGA configuration (#54)
Include	Platform: introduce BootApp for hotkey management
Library	Platform: introduce BootApp for hotkey management
PrePi	Project: Update to EDK2 master (#61)
Resources	UC120: stub SPI config file
Tools	Project: Update to EDK2 master (#61)
.clang-format	Standarized code styles (clang-format) (#37)
.gitignore	ACPI: Build SSDT(s) at compile time.
FvWrapper.ld	Port DB820c target to 950XL.
Hapanero.dsc	Fix for overwritten values in platform definitions (#26)
Hapanero.fdf	BootApp: clear prompt before booting apps
LICENSE	Project: Update LICENSE.
Lumia950.dsc	talkman: Fixing a typo about the processor model (#40)
Lumia950.fdf	BootApp: clear prompt before booting apps
Lumia950XL.dsc	Fix for overwritten values in platform definitions (#26)
Lumia950XL.fdf	BootApp: clear prompt before booting apps
Lumia950XL.fdf.inc	Build: ready for next public release.
Lumia950XLPkg.dec	Platform: introduce BootApp for hotkey management
README.md	Project: update README.md [skip ci]

Lumia950XLPkg status

- Tracking the TianoCore master branch
- SoC devices
  - Some drivers are ported from LK and the EFIDroid project
  - Power Management (PMIC, RPM, Pinctrl)
  - Low speed I/O (GPIO, I2C, SPI)
  - High speed I/O (SDHCI, PCIe)
  - FrameBuffer display
- Peripherals
  - Synaptics RMI4 I2C digitizer
  - Lattice iCE5LP2K bitstream uploader
- ACPI ready

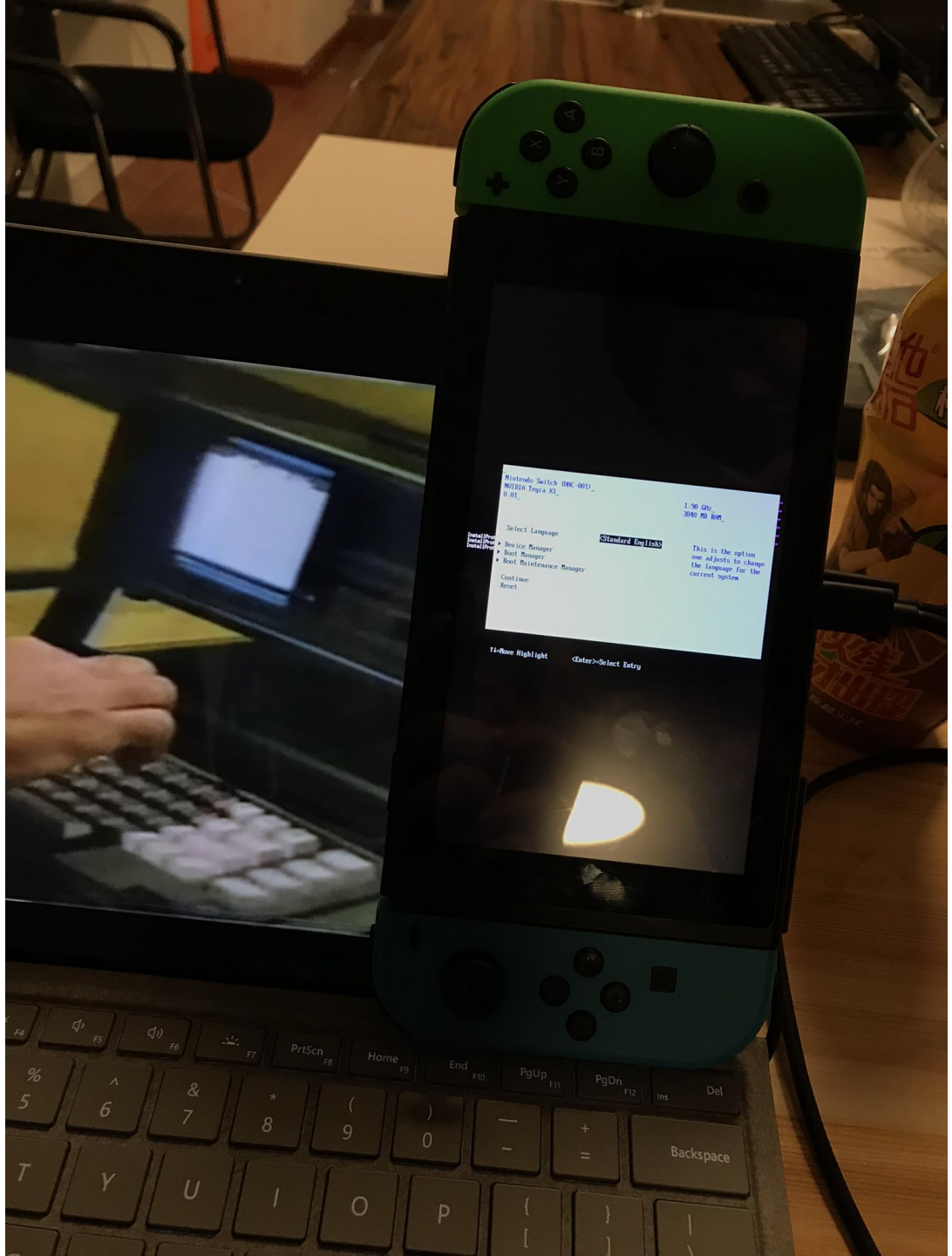
Bonus: Nintendo Switch

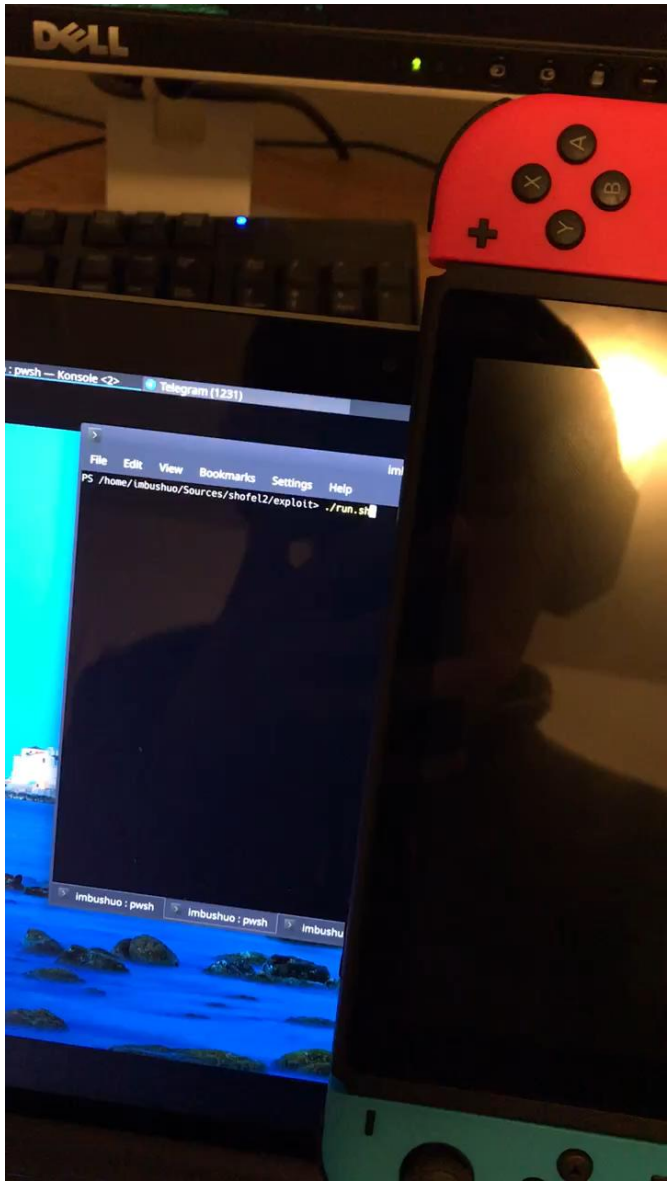
Tweet

James Swineson
@zhjits

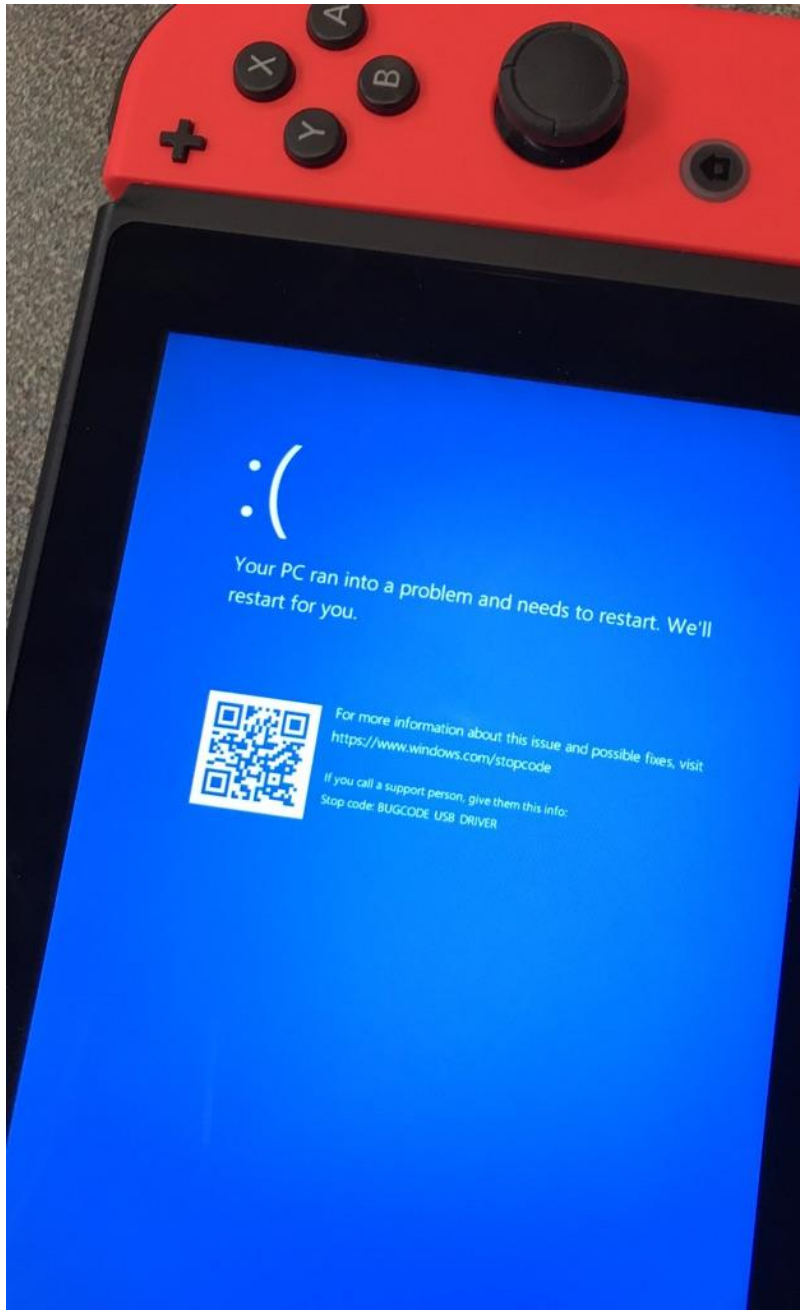
@imbushuo is playing my Nintendo Switch. Guess what will happen next.

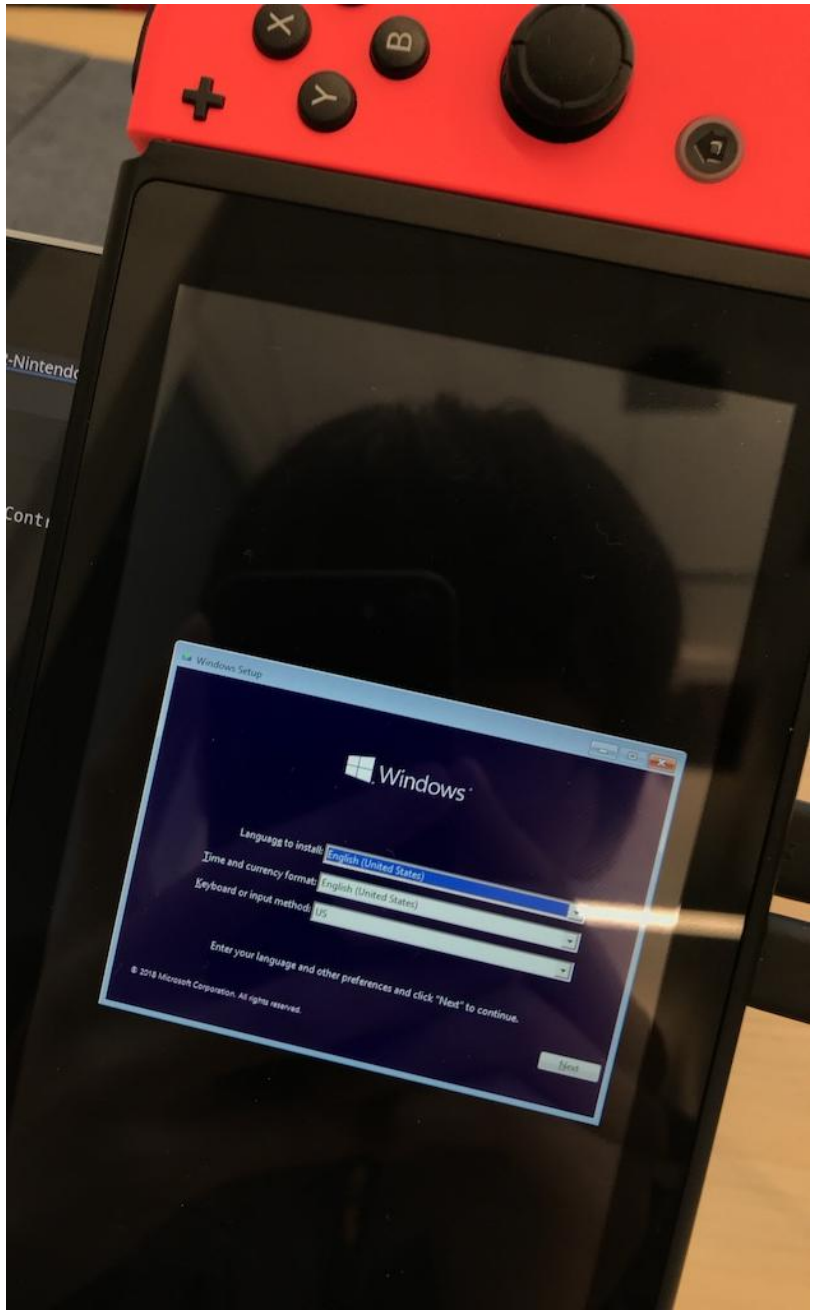
4:03 AM · May 14, 2018 · Twitter for iPhone

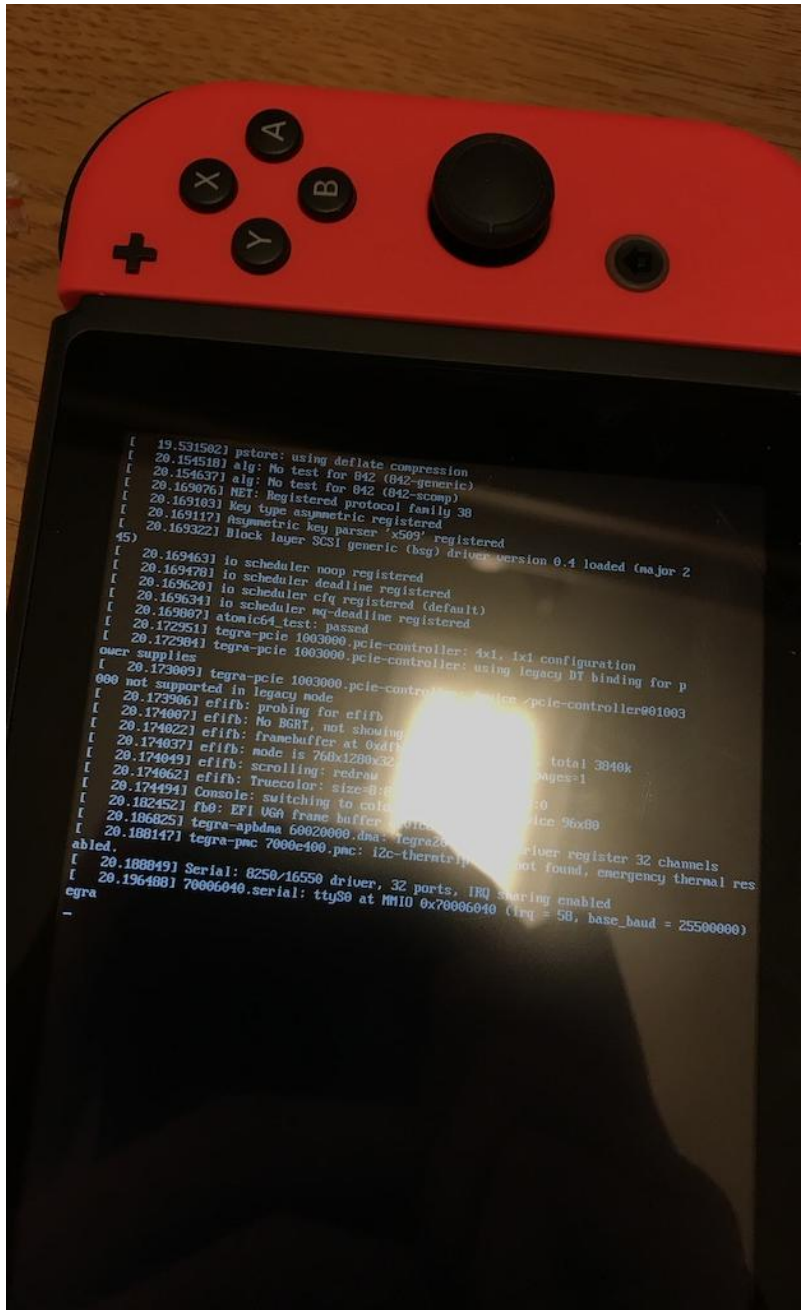

Nintendo Switch (HAC-001)
NVIDIA Tegra X1
0.01
1.90 GHz
3040 MB RAM
Select Language
<Standard English>
Device Manager
Boot Manager
Boot Maintenance Manager
Continue
Reset
This is the option one adjusts to change the language for the current system
↑↓=Move Highlight
<Enter>=Select Entry



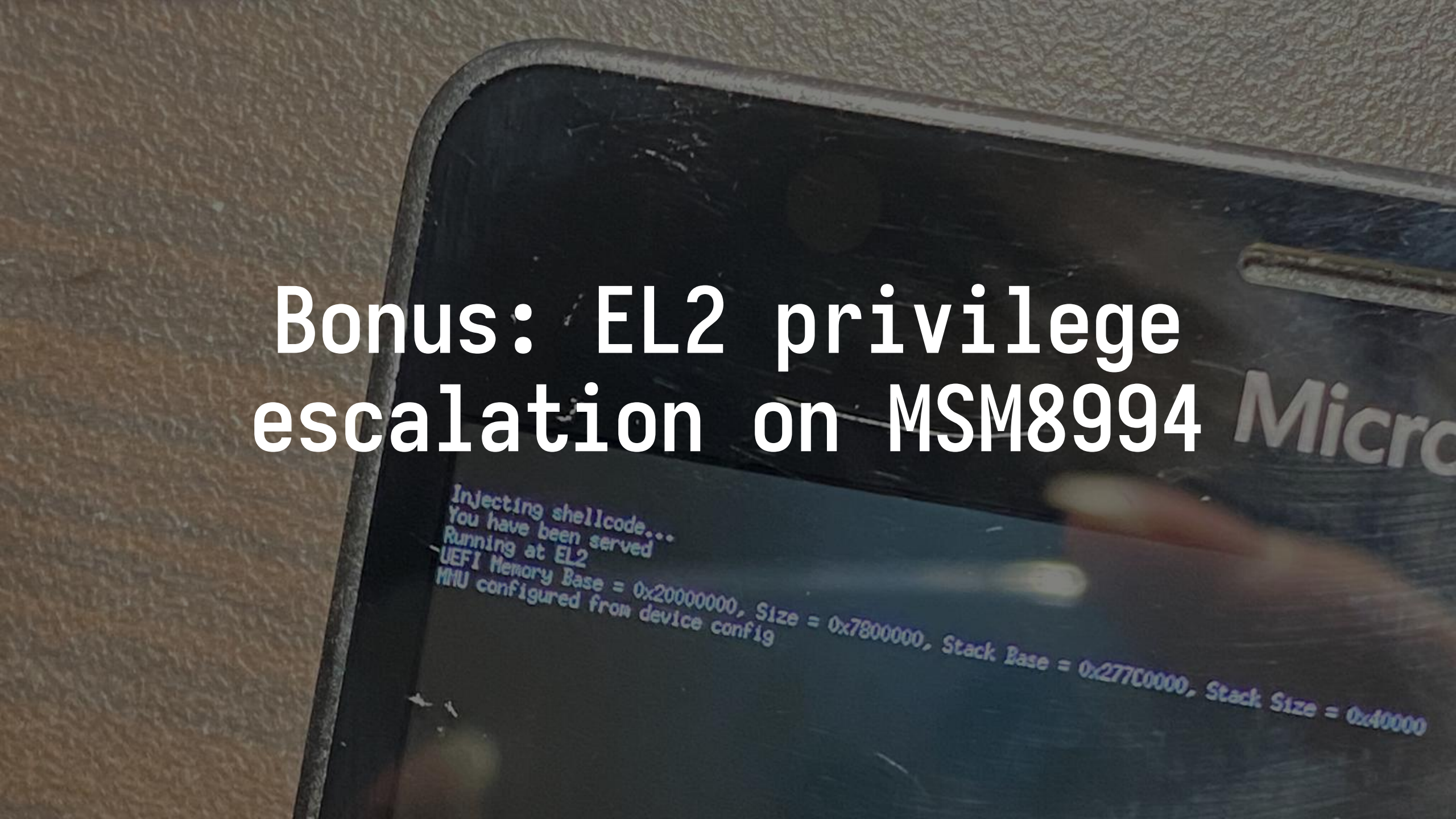
https://www.youtube.com/watch?v=FyjPGOj0tiQ


:(
Your PC ran into a problem and needs to restart. We'll restart for you.
For more information about this issue and possible fixes, visit https://www.windows.com/stopcode
If you call a support person, give them this info:
Stop code: BUGCODE_USB_DRIVER


Windows Setup
Windows
Language to install: English (United States)
Time and currency format: English (United States)
Keyboard or input method: US
Enter your language and other preferences and click "Next" to continue.
© 2018 Microsoft Corporation. All rights reserved.
Next


[19.531502] pstore: using deflate compression
[20.154518] alg: No test for 842 (842-generic)
[20.154637] alg: No test for 842 (842-scomp)
[20.169076] NET: Registered protocol family 38
[20.169103] Key type asymmetric registered
[20.169117] Asymmetric key parser 'x509' registered
[20.169322] Block layer SCSI generic (bsg) driver version 0.4 loaded (major 2
45)
[20.169463] io scheduler noop registered
[20.169478] io scheduler deadline registered
[20.169620] io scheduler cfq registered (default)
[20.169634] io scheduler mq-deadline registered
[20.169807] atomic64_test: passed
[20.172951] tegra-pcie 1003000.pcie-controller: 4x1, 1x1 configuration
[20.172984] tegra-pcie 1003000.pcie-controller: using legacy DT binding for p
ower supplies
[20.173009] tegra-pcie 1003000.pcie-controller: /pcie-controller@01003
000 not supported in legacy mode
[20.173906] efifb: probing for efifb
[20.174007] efifb: No BGRT, not showing
[20.174022] efifb: framebuffer at 0xdf
[20.174037] efifb: mode is 768x1280x32 , total 3040k
[20.174049] efifb: scrolling: redraw , pages=1
[20.174062] efifb: truecolor: size=8:
[20.174494] Console: switching to colo
[20.182452] fb0: EFI VGA frame buffer
[20.186825] tegra-apbdma 60020000.dma: Tegra2
[20.188147] tegra-pmc 7000e400.pmc: i2c-thermtrip
abled.
[20.188849] Serial: 8250/16550 driver, 32 ports, IRQ sharing enabled
[20.196488] 70006040.serial: ttyS0 at MMIO 0x70006040 (irq = 58, base_baud = 25500000)
egra

Bonus: EL2 privilege escalation on MSM8994



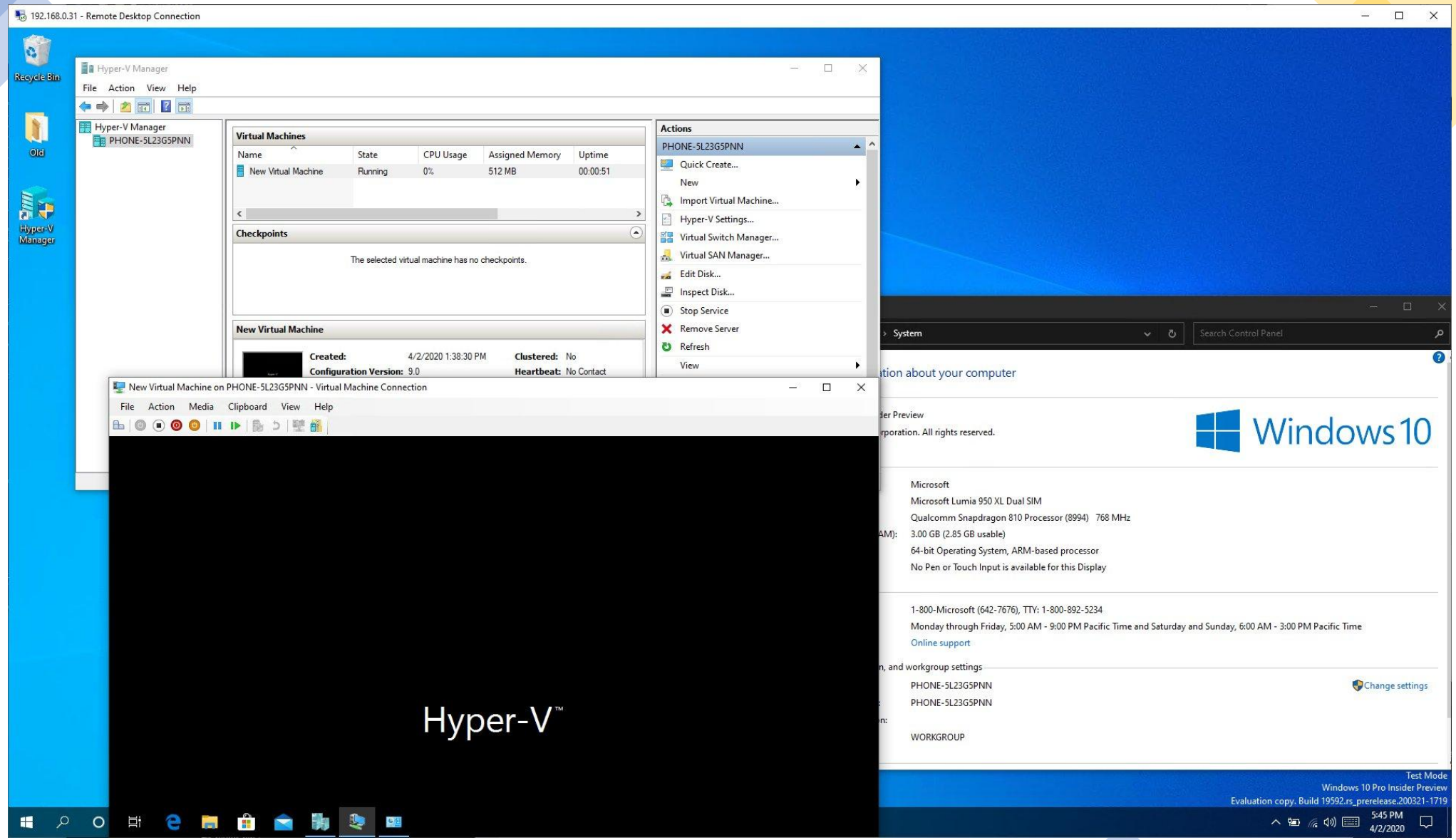
```
// but if secondary CPUs are launched, exception vector will be
// fixed so CPU0 call would fail. Therefore we patched
// PSCI_CPU_SUSPEND_AARCH64 handler at 0x06c03aa8.
DEBUG((EFI_D_ERROR, "Injecting shellcode...\n"));
EFI_PHYSICAL_ADDRESS PsciCpuSuspendHandlerAddr = 0x06c03aa8;
UINT8 *PsciCpuSuspendHandler = (UINT8 *)(VOID *)PsciCpuSuspendHandlerAddr;
CopyMem(PsciCpuSuspendHandler, El2ShellCode, sizeof(El2ShellCode));
ArmDataSynchronizationBarrier();
ArmInvalidateDataCache();
```

EL2 privilege escalation

- On MSM8992 and MSM8994, EL2 hypervisor resides in the trust boundary of EL1 supervisor (aka. kernel)
- Finding the exception vector…address writable
- Let's patch the vector table on the fly
- https://www.blackhat.com/docs/us-17/wednesday/us-17-Bazhaniuk-BluePill-For-Your-Phone.pdf

```
30
31  // ArmDisableInterrupts
32  msr   daifset, #DAIF_WR_INT_BITS
33  isb
34
35  // ArmDisableCachesAndMmu
36     EL1_OR_EL2_OR_EL3(x1)
37  1: mrs     x0, sctlr_el1        // Get control register EL1
38     b       4f
39  2: mrs     x0, sctlr_el2        // Get control register EL2
40     b       4f
41  3: mrs     x0, sctlr_el3        // Get control register EL3
42  4: mov     x1, #~(CTRL_M_BIT | CTRL_C_BIT | CTRL_I_BIT)  // Disabl
43     and     x0, x0, x1
44     EL1_OR_EL2_OR_EL3(x1)
45  1: msr     sctlr_el1, x0        // Write back control register
46     b       4f
47  2: msr     sctlr_el2, x0        // Write back control register
48     b       4f
49  3: msr     sctlr_el3, x0        // Write back control register
50  4: dsb     sy
51     isb
52
53  // Load address and branch
54  mov x0, #0x00200000
55  br x0       You, 2 months ago • EL2: initial EL2 exploit experiment
```

```
// Install patch
InstallEl2Patch();

// Looks good. Notify all secondary CPUs to jump!
for (UINTN Index = 1; Index < FixedPcdGet32(PcdCoreCount); Index++) {
  EFI_PHYSICAL_ADDRESS MailboxAddress =
      FixedPcdGet64(SecondaryCpuMpParkRegionBase) + 0x10000 * Index +
      0x1000;
  PEFI_PROCESSOR_MAILBOX pMailbox =
      (PEFI_PROCESSOR_MAILBOX)(VOID *)MailboxAddress;

  pMailbox->El2JumpFlag = EL2REDIR_MAILBOX_SIGNAL;
  ArmDataSynchronizationBarrier();
}

// Make sure they are all initialized
DEBUG((EFI_D_ERROR, "Waiting for all CPUs...\n"));
WaitForSecondaryCPUs();
DEBUG((EFI_D_ERROR, "All CPU started.\n"));
ArmDataSynchronizationBarrier();

DEBUG((EFI_D_ERROR, "Jump CPU0 to EL2.\n"));
ArmDataSynchronizationBarrier();

// Install patch again
InstallEl2Patch();

// Jump overself
ARM_HVC_ARGS StubArg;
// PSCI_CPU_SUSPEND_AA64
StubArg.Arg0 = 0xc4000001;
ArmCallHvc(&StubArg);
```


192.168.0.31 - Remote Desktop Connection
Recycle Bin
Old
Hyper-V Manager
Hyper-V Manager
File Action View Help
Hyper-V Manager
PHONE-5L23G5PNN
Virtual Machines
Name State CPU Usage Assigned Memory Uptime
New Virtual Machine Running 0% 512 MB 00:00:51
Checkpoints
The selected virtual machine has no checkpoints.
New Virtual Machine
Created: 4/2/2020 1:38:30 PM
Clustered: No
Configuration Version: 9.0
Heartbeat: No Contact
Actions
PHONE-5L23G5PNN
Quick Create...
New
Import Virtual Machine...
Hyper-V Settings...
Virtual Switch Manager...
Virtual SAN Manager...
Edit Disk...
Inspect Disk...
Stop Service
Remove Server
Refresh
View
New Virtual Machine on PHONE-5L23G5PNN - Virtual Machine Connection
File Action Media Clipboard View Help
Hyper-V™
System
Search Control Panel
ition about your computer
Windows 10
Microsoft
Microsoft Lumia 950 XL Dual SIM
Qualcomm Snapdragon 810 Processor (8994) 768 MHz
3.00 GB (2.85 GB usable)
64-bit Operating System, ARM-based processor
No Pen or Touch Input is available for this Display
1-800-Microsoft (642-7676), TTY: 1-800-892-5234
Monday through Friday, 5:00 AM - 9:00 PM Pacific Time and Saturday and Sunday, 6:00 AM - 3:00 PM Pacific Time
Online support
PHONE-5L23G5PNN
PHONE-5L23G5PNN
Change settings
WORKGROUP
Test Mode
Windows 10 Pro Insider Preview
Evaluation copy. Build 19592.rs_prerelease.200321-1719
5:45 PM
4/2/2020