Linux Kernel Overview

Nick Kossifidis - HY428 12/5/2023

History and features

What is Linux...

- Monolithic kernel
- Built in 1991 by Linus Torvalds
- Written in C
- Licensed under the GPLv2
- Open standards (e.g. POSIX / SUS) compliant
- Multi-arch
- Multi-user
- Preemptive multitasking
- Scalable
- Fully Customizable
- Bleeding edge !

Linux is huge !

~23 million lines of code and counting

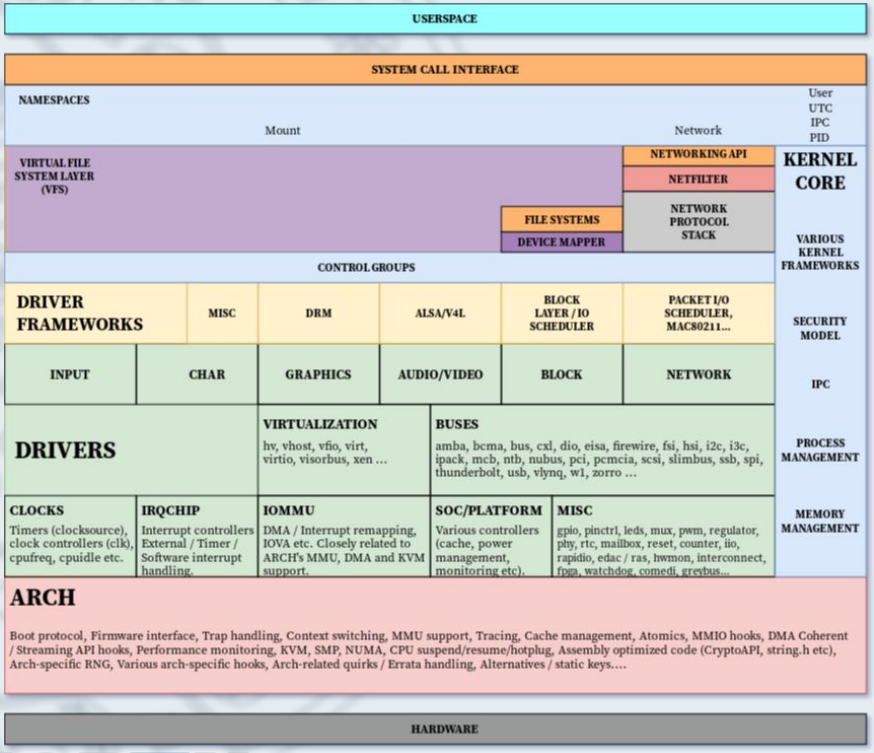
~64k files in the kernel tree

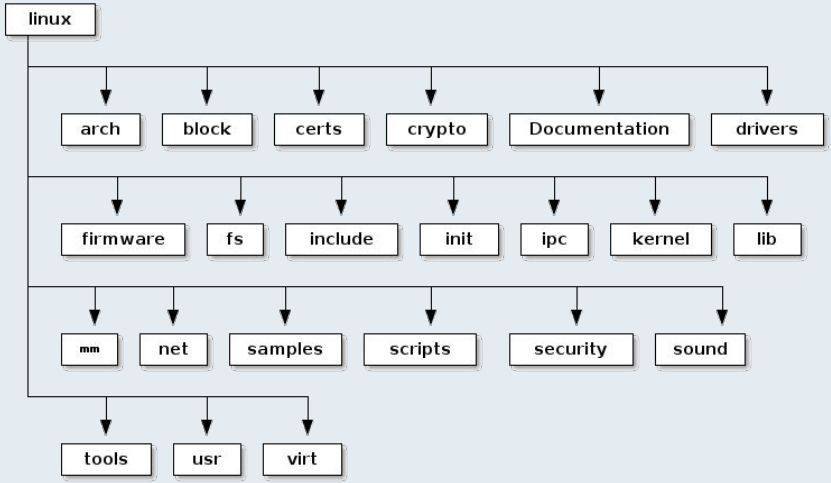
Almost 2.000 developers / release

More than 20.000 developers have contributed so far

Under the hood

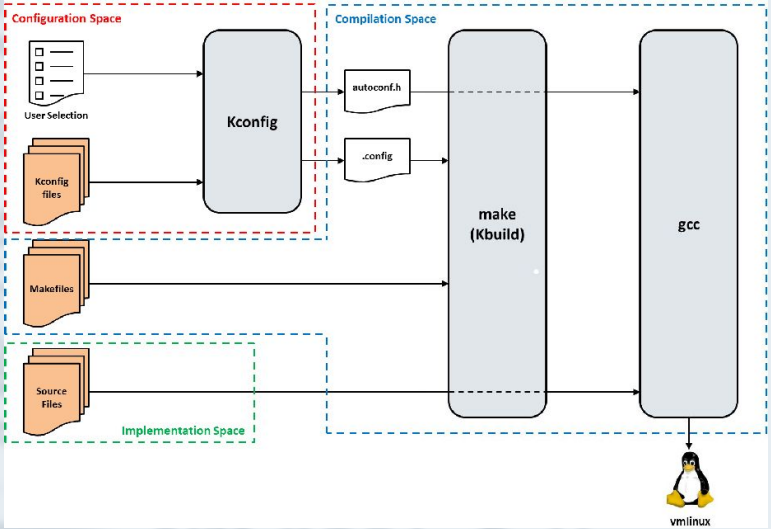
A visualisation attempt

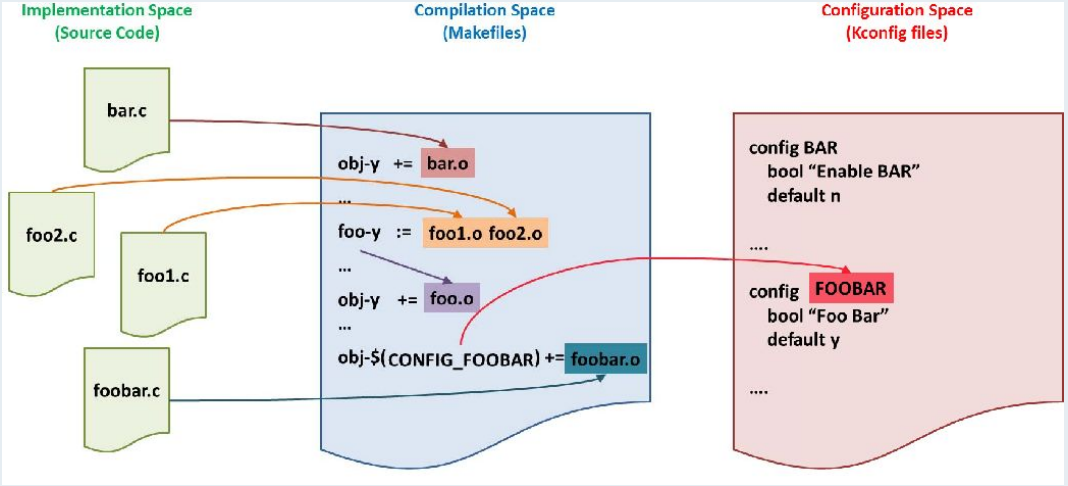




FORTH
INSTITUTE OF COMPUTER SCIENCE

Kernel Build System






```
#ifdef CONFIG_FOOBAR /* Or #if IS_ENABLED(CONFIG_FOOBAR) */
...
#endif
```

FORTH
INSTITUTE OF COMPUTER SCIENCE

Linux kernel modules

- Object files that extend the kernel's functionality.
- Can be compiled as part of the kernel (built-in) or loaded at runtime (loadable).
- Each module has a module_init function that runs when the module is added to the kernel.
- For built-in modules module_init translates to a device initcall so that it becomes part of the initcall chain.
- Loadable modules in contrast with built-in modules can be unloaded and can provide a module_exit function to do the cleanup.
- Modules can register a set of module parameters using the module_param macro. Parameters can be passed on loadable modules via the insmod/modprobe command and to built-in modules via the kernel's cmdline in the form module_name.parameter=value.
- They can also contain information such as the module's author, license, description and parameter description. To access that info use the modinfo command.
- Modules support versioning, can contain a checksum for further validation and can be protected by signature checks on recent kernels so that only signed modules can be loaded.

Some extra kernel <-> user interfaces

- Pseudo file systems (VFS)
- /dev -> Device access
- /proc -> Runtime kernel / process status/parameters
  - /proc/self/ or /proc/$pid -> Process state/config
  - /proc/cpuinfo -> CPU layout
  - /proc/vmstat -> Virtual Memory statistics
  - /proc/interrupts -> Interrupt counters
  - /proc/cmdline -> Kernel command line
  - /proc/sys -> System status/parameters
  - …
- /sys -> System layout (device model)
  - /sys/block -> Block devices
  - /sys/bus -> Registered buses
  - /sys/class -> Devices by device class
  - /sys/devices -> Devices layout
  - /sys/kernel -> Runtime kernel parameters, includes configfs and debugfs mountpoints
  - /sys/modules -> Loaded kernel modules and their parameters (under the parameters/ subfolder)
  - …
- Sysctl/ioctl/netlink
- Device-mapper uevents
- Kdbus (IPC transport)
- CryptoAPI
- …
- Vdso

Some companion libraries/daemons

- Libaio -> Advanced I/O
- Libdrm -> Direct Rendering Manager
- Libmesa -> Graphics Acceleration
- Libasound -> Access to sound devices via ALSA
- Libevdev -> Access to event interface (/dev/input)
- Libnl -> Generic Netlink
- Libnftln -> NFTables (Netfilter configuration)
- Iproute2 -> Network configuration
- Cfg80211/nl80211 -> WiFi configuration / driver access
- …

- Udev -> Userspace /dev (hotplugging, firmware loading etc)
- Pulseaudio/JackD/Pipewire -> Access to sound cards (multiplexing, mixing etc)
- Irqbalance -> IRQ balancing across CPUs
- Wpa supplicant / xsupplicant -> 802.11i (WPA) 802.1x (PNAC / MacSEC)
- Hostapd -> WiFi Access Pont daemon
- …

Firmware services...

- CPU-level power management (cpu hotplug)
- Platform-level power management (suspend / resume)
- Early access to console / Framebuffer
- Performance monitoring
- Clock / timer setup
- Initial physical memory map setup
- Various privileged operations, including interaction with the Secure Monitor etc
- …
- UEFI Boot/Runtime services:
  - https://uefi.org/specs/UEFI/2.9_A/07_Services_Boot_Services.html
  - https://uefi.org/specs/UEFI/2.9_A/08_Services_Runtime_Services.html
- ARM Trusted firmware:
  - https://elinux.org/images/0/05/Elc-tfa.pdf
- RISC-V SBI:
  - https://github.com/riscv-non-isa/riscv-sbi-doc/blob/master/riscv-sbi.pdf

Open source firmware implementations:

Tianocore / EDK2: https://www.tianocore.org/ (UEFI)

CoreBoot: https://www.coreboot.org/

OpenSBI: https://github.com/riscv-software-src/opensbi (RISC-V)

ARM Trusted Firmware: https://github.com/ARM-software/arm-trusted-firmware (ARM)

Device discovery…

- Desktop Management Interface (DMI) tables (SMBIOS)
  - https://www.dmtf.org/standards/dmi
- Advanced Control Power Interface (ACPI) tables
  - https://uefi.org/htmlspecs/ACPI_Spec_6_4_html/index.html
- Device-tree
  - https://www.devicetree.org/

devicetree
.org

Data available through /sys/firmware/

Decode with dmidecode / acpidump & iasl / fdtdump

Other useful tools: lspci, lsusb, lshw, i2cdetect…

Device discovery…

```
kazofonias /home/mick # dmidecode
# dmidecode 3.4
Getting SMBIOS data from sysfs.
SMBIOS 3.2.0 present.
Table at 0xBD7F3000.

Handle 0x0000, DMI type 0, 26 bytes
BIOS Information
        Vendor: American Megatrends Inc.
        Version: 8701
        Release Date: 02/08/2023
        Address: 0xF0000
        Runtime Size: 64 kB
        ROM Size: 16 MB
        Characteristics:
                PCI is supported
                APM is supported
                BIOS is upgradeable
                BIOS shadowing is allowed
                Boot from CD is supported
                Selectable boot is supported
                BIOS ROM is socketed
                EDD is supported
                5.25"/1.2 MB floppy services are supported (int 13h)
                3.5"/720 kB floppy services are supported (int 13h)
                3.5"/2.88 MB floppy services are supported (int 13h)
                Print screen service is supported (int 5h)
                8042 keyboard services are supported (int 9h)
                Serial services are supported (int 14h)
                Printer services are supported (int 17h)
                ACPI is supported
                USB legacy is supported
                BIOS boot specification is supported
                Targeted content distribution is supported
                UEFI is supported
        BIOS Revision: 5.17

Handle 0x0001, DMI type 1, 27 bytes
System Information
        Manufacturer: ASUS
        Product Name: System Product Name
        Version: System Version
        Serial Number: System Serial Number
        UUID: bceedb20-24c3-11e7-90ef-6045cb9e89a7
        Wake-up Type: Power Switch
        SKU Number: SKU
        Family: To be filled by O.E.M.

Handle 0x0002, DMI type 2, 15 bytes
Base Board Information
        Manufacturer: ASUSTeK COMPUTER INC.
        Product Name: CROSSHAIR VI HERO
        Version: Rev 1.xx
        Serial Number: 170499331103078
        Asset Tag: Default string
        Features:
                Board is a hosting board
                Board is replaceable
        Location In Chassis: Default string
        Chassis Handle: 0x0003
        Type: Motherboard
        Contained Object Handles: 0
```

```
Handle 0x0034, DMI type 4, 48 bytes
Processor Information
        Socket Designation: AM4
        Type: Central Processor
        Family: Zen
        Manufacturer: Advanced Micro Devices, Inc.
        ID: 12 0F A2 00 FF FB 8B 17
        Signature: Family 25, Model 33, Stepping 2
        Flags:
                FPU (Floating-point unit on-chip)
                VME (Virtual mode extension)
                DE (Debugging extension)
                PSE (Page size extension)
                TSC (Time stamp counter)
                MSR (Model specific registers)
                PAE (Physical address extension)
                MCE (Machine check exception)
                CX8 (CMPXCHG8 instruction supported)
                APIC (On-chip APIC hardware supported)
                SEP (Fast system call)
                MTRR (Memory type range registers)
                PGE (Page global enable)
                MCA (Machine check architecture)
                CMOV (Conditional move instruction supported)
                PAT (Page attribute table)
                PSE-36 (36-bit page size extension)
                CLFSH (CLFLUSH instruction supported)
                MMX (MMX technology supported)
                FXSR (FXSAVE and FXSTOR instructions supported)
                SSE (Streaming SIMD extensions)
                SSE2 (Streaming SIMD extensions 2)
                HTT (Multi-threading)
        Version: AMD Ryzen 7 5700X 8-Core Processor
        Voltage: 1.1 V
        External Clock: 100 MHz
        Max Speed: 4650 MHz
        Current Speed: 3400 MHz
        Status: Populated, Enabled
        Upgrade: Socket AM4
        L1 Cache Handle: 0x0031
        L2 Cache Handle: 0x0032
        L3 Cache Handle: 0x0033
        Serial Number: Unknown
        Asset Tag: Unknown
        Part Number: Unknown
        Core Count: 8
        Core Enabled: 8
        Thread Count: 16
        Characteristics:
                64-bit capable
                Multi-Core
                Hardware Thread
                Execute Protection
                Enhanced Virtualization
                Power/Performance Control
```

```
Handle 0x0038, DMI type 17, 84 bytes
Memory Device
        Array Handle: 0x002E
        Error Information Handle: 0x0037
        Total Width: 64 bits
        Data Width: 64 bits
        Size: 8 GB
        Form Factor: DIMM
        Set: None
        Locator: DIMM_A2
        Bank Locator: BANK 1
        Type: DDR4
        Type Detail: Synchronous Unbuffered (Unregistered)
        Speed: 3200 MT/s
        Manufacturer: G Skill Intl
        Serial Number: 00000000
        Asset Tag: Not Specified
        Part Number: F4-3200C14-8GTZ
        Rank: 1
        Configured Memory Speed: 3200 MT/s
        Minimum Voltage: 1.2 V
        Maximum Voltage: 1.2 V
        Configured Voltage: 1.2 V
        Memory Technology: DRAM
        Memory Operating Mode Capability: Volatile memory
        Firmware Version: Unknown
        Module Manufacturer ID: Bank 5, Hex 0xCD
        Module Product ID: Unknown
        Memory Subsystem Controller Manufacturer ID: Unknown
        Memory Subsystem Controller Product ID: Unknown
        Non-Volatile Size: None
        Volatile Size: 8 GB
        Cache Size: None
        Logical Size: None

Handle 0x0039, DMI type 20, 35 bytes
Memory Device Mapped Address
        Starting Address: 0x00000000000
        Ending Address: 0x003FFFFFFFF
        Range Size: 16 GB
        Physical Device Handle: 0x0038
        Memory Array Mapped Address Handle: 0x0030
        Partition Row Position: Unknown
        Interleave Position: Unknown
        Interleaved Data Depth: Unknown
```

DMTF®

FORTH
INSTITUTE OF COMPUTER SCIENCE

Device discovery...

```
SSDT @ 0x0000000000000000
    0000: 53 53 44 54 AF 10 00 00 02 EB 41 4D 44 00 00 00  SSDT......AMD...
    0010: 4D 59 52 54 4C 45 00 00 01 00 00 00 49 4E 54 4C  MYRTLE......INTL
    0020: 13 09 12 20 08 4D 57 54 54 0A FF 10 40 42 5C 2E  ... .MWTT...@B\.
    0030: 5F 53 42 5F 49 32 43 41 5B 82 47 07 57 54 31 41  _SB_I2CA[.G.WTTA
    0040: 08 5F 41 44 52 00 08 5F 48 49 44 0D 53 54 4B 30  ._ADR.._HID.STK0
    0050: 30 30 31 41 00 08 5F 43 49 44 0D 53 50 42 54 65  001A.._CID.SPBTe
    0060: 73 74 4B 4D 44 46 41 00 14 32 5F 43 52 53 00 08  stKMDFA..2_CRS..
    0070: 52 42 55 46 11 21 0A 1E 8E 19 00 01 00 01 02 00  RBUF.!..........

MCFG @ 0x0000000000000000
    0000: 4D 43 46 47 3C 00 00 00 01 F4 41 4C 41 53 4B 41  MCFG<.....ALASKA
    0010: 41 20 4D 20 49 00 00 00 09 20 07 01 4D 53 46 54  A M I.... ..MSFT
    0020: 13 00 01 00 00 00 00 00 00 00 00 00 00 00 00 F0  ................
    0030: 00 00 00 00 00 00 00 7F 00 00 00 00              ............

APIC @ 0x0000000000000000
    0000: 41 50 49 43 5E 01 00 00 04 DE 41 4C 41 53 4B 41  APIC^.....ALASKA
    0010: 41 20 4D 20 49 20 00 00 09 20 07 01 41 4D 49 20  A M I .... ..AMI
    0020: 13 00 01 00 00 00 E0 FE 01 00 00 00 08 00 00     ...............


CRAT @ 0x0000000000000000
    0000: 43 52 41 54 10 0F 00 00 01 ED 41 4D 44 00 00 00  CRAT......AMD...
    0010: 41 6D 64 54 61 62 6C 65 01 00 00 00 41 4D 44 20  AmdTable....AMD
    0020: 01 00 00 00 3D 00 00 00 01 00 00 00 00 00 00 00  ....=...........
    0030: 00 28 00 00 05 00 00 00 00 00 00 00 00 00 00 00  .(..............
    0040: 10 00 00 00 00 00 00 00 00 00 00 00 04 00 00 00  ................

PCCT @ 0x0000000000000000
    0000: 50 43 43 54 6E 00 00 00 02 B9 41 4D 44 00 00 00  PCCTn.....AMD...
    0010: 41 6D 64 54 61 62 6C 65 01 00 00 00 41 4D 44 20  AmdTable....AMD
    0020: 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................

SSDT @ 0x0000000000000000
    0000: 53 53 44 54 B6 3C 00 00 02 A7 41 4D 44 00 00 00  SSDT.<....AMD...
    0010: 41 4D 44 20 41 4F 44 00 01 00 00 00 49 4E 54 4C  AMD AOD.....INTL
    0020: 13 09 12 20 10 81 C9 03 5C 00 5B 82 8A C8 03 41  ... ....\.[....A
    0030: 4F 44 5F 08 44 42 47 5F 00 08 47 44 42 47 00 08  OD_.DBG_..GDBG..
    0040: 43 44 42 47 00 08 56 44 42 47 00 08 52 44 42 47  CDBG..VDBG..RDBG
    0050: 00 5B 80 50 4D 52 47 00 0C 00 03 D8 FE 0B 00 01  .[.PMRG.........
    0060: 5B 80 50 53 4D 49 01 0A B2 0A 02 5B 81 10 50 53  [.PSMI.....[..PS
    0070: 4D 49 01 41 50 4D 43 08 41 50 4D 44 08 14 10 41  MI.APMC.APMD...A
    0080: 53 4D 49 01 70 68 41 50 4D 43 5B 22 0A 0A 08 4F  SMI.phAPMC["...O
    0090: 42 49 44 12 46 53 C1 0A 4D 0C 01 00 01 00 0C 02  BID.FS..M.......
    00A0: 00 01 00 0C 01 00 02 00 0C 02 00 02 00 0C 03 00  ................

TPM2 @ 0x0000000000000000
    0000: 54 50 4D 32 4C 00 00 00 04 14 41 4C 41 53 4B 41  TPM2L.....ALASKA
    0010: 41 20 4D 20 49 20 00 00 01 00 00 00 41 4D 49 20  A M I ......AMI
    0020: 00 00 00 00 00 00 00 00 10 05 21 FD 00 00 00 00  ..........!.....
    0030: 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
    0040: 00 00 01 00 00 10 9B BC 00 00 00 00              ............

VFCT @ 0x0000000000000000
    0000: 56 46 43 54 84 B2 00 00 01 BE 41 4C 41 53 4B 41  VFCT......ALASKA
    0010: 41 20 4D 20 49 20 00 00 01 00 00 00 41 4D 44 00  A M I ......AMD.
    0020: 47 4F 50 31 32 9B A3 5D BD C6 CF 49 95 A6 E8 E4  GOP12..]...I....
    0030: 2E CD 79 A7 4C 00 00 00 00 00 00 00 00 00 00 00  ..y.L...........
    0040: 00 00 00 00 00 00 00 00 00 00 00 00 0D 00 00 00  ................

CDIT @ 0x0000000000000000
    0000: 43 44 49 54 29 00 00 00 01 E7 41 4D 44 00 00 00  CDIT).....AMD...
    0010: 41 6D 64 54 61 62 6C 65 01 00 00 00 41 4D 44 20  AmdTable....AMD
    0020: 01 00 00 00 01 00 00 00 0A                       .........

IVRS @ 0x0000000000000000
    0000: 49 56 52 53 D0 00 00 00 02 D2 41 4D 44 20 20 00  IVRS......AMD  .
    0010: 41 6D 64 54 61 62 6C 65 01 00 00 00 41 4D 44 20  AmdTable....AMD
    0020: 01 00 00 00 30 20 00 00 00 00 00 00 00 00 00 00  ....0 ..........
    0030: 10 B0 48 00 02 00 40 00 00 00 50 FD 00 00 00 00  ..H...@...P.....
```

```
Gazofonias /tmp/acpidump # acpidump -b
Gazofonias /tmp/acpidump # ls
apic.dat  bgrt.dat  cdit.dat  crat.dat  dsdt.dat  facp.dat  facs.dat  fidt.dat  fpdt.d
Gazofonias /tmp/acpidump # iasl * &> /dev/null
Gazofonias /tmp/acpidump # ls
apic.dat  bgrt.dat  cdit.dat  crat.dat  dsdt.dat  facp.dat  facs.dat  fidt.dat  fpdt.d
apic.dsl  bgrt.dsl  cdit.dsl  crat.dsl  dsdt.dsl  facp.dsl  facs.dsl  fidt.dsl  fpdt.d
Gazofonias /tmp/acpidump # cat ssdt3.dsl
/*
 * Intel ACPI Component Architecture
 * AML/ASL+ Disassembler version 20200717 (64-bit version)
 * Copyright (c) 2000 - 2020 Intel Corporation
 *
 * Disassembling to symbolic ASL+ operators
 *
 * Disassembly of ssdt3.dat, Wed May 17 13:12:03 2023
 *
 * Original Table Header:
 *     Signature        "SSDT"
 *     Length           0x000000FC (252)
 *     Revision         0x02
 *     Checksum         0xBE
 *     OEM ID           "ALASKA"
 *     OEM Table ID     "CPUSSDT"
 *     OEM Revision     0x01072009 (17244169)
 *     Compiler ID      "AMI "
 *     Compiler Version 0x01072009 (17244169)
 */
DefinitionBlock ("", "SSDT", 2, "ALASKA", "CPUSSDT", 0x01072009)
{
    Scope (\_PR)
    {
        Processor (C000, 0x00, 0x00000810, 0x06){}
        Processor (C001, 0x01, 0x00000810, 0x06){}
        Processor (C002, 0x02, 0x00000810, 0x06){}
        Processor (C003, 0x03, 0x00000810, 0x06){}
        Processor (C004, 0x04, 0x00000810, 0x06){}
        Processor (C005, 0x05, 0x00000810, 0x06){}
        Processor (C006, 0x06, 0x00000810, 0x06){}
        Processor (C007, 0x07, 0x00000810, 0x06){}
        Processor (C008, 0x08, 0x00000810, 0x06){}
        Processor (C009, 0x09, 0x00000810, 0x06){}
        Processor (C00A, 0x0A, 0x00000810, 0x06){}
        Processor (C00B, 0x0B, 0x00000810, 0x06){}
        Processor (C00C, 0x0C, 0x00000810, 0x06){}
        Processor (C00D, 0x0D, 0x00000810, 0x06){}
        Processor (C00E, 0x0E, 0x00000810, 0x06){}
        Processor (C00F, 0x0F, 0x00000810, 0x06){}
    }
}
```

FORTH
INSTITUTE OF COMPUTER SCIENCE

Device discovery...

```
sazofotlas /tmp # fdtdump qemu-riscv.dtb

**** fdtdump is a low-level debugging tool, not meant for general use.
**** If you want to decompile a dtb, you probably want
****     dtc -I dtb -O dts <filename>

/dts-v1/;
// magic:               0xd00dfeed
// totalsize:           0x14be (5310)
// off_dt_struct:       0x38
// off_dt_strings:      0x131c
// off_mem_rsvmap:      0x28
// version:             17
// last_comp_version:   2
// boot_cpuid_phys:     0x0
// size_dt_strings:     0x1a2
// size_dt_struct:      0x12e4

/ {
    #address-cells = <0x00000002>;
    #size-cells = <0x00000002>;
    compatible = "riscv-virtio";
    model = "riscv-virtio,qemu";
    fw-cfg@10100000 {
        dma-coherent;
        reg = <0x00000000 0x10100000 0x00000000 0x00000018>;
        compatible = "qemu,fw-cfg-mmio";
    };
    flash@20000000 {
        bank-width = <0x00000004>;
        reg = <0x00000000 0x20000000 0x00000000 0x02000000 0x00000000 0x22000000 0x00000000 0x02000000>;
        compatible = "cfi-flash";
    };
    chosen {
        linux,initrd-end = <0x885c2e40>;
        linux,initrd-start = <0x88200000>;
        stdout-path = "/soc/uart@10000000";
    };
    platform@4000000 {
        interrupt-parent = <0x00000009>;
        ranges = <0x00000000 0x00000000 0x04000000 0x02000000>;
        #address-cells = <0x00000001>;
        #size-cells = <0x00000001>;
        compatible = "qemu,platform", "simple-bus";
    };
    memory@80000000 {
        device_type = "memory";
        reg = <0x00000000 0x80000000 0x00000001 0x00000000>;
    };
    cpus {
        #address-cells = <0x00000001>;
        #size-cells = <0x00000000>;
        timebase-frequency = <0x00989680>;
        cpu@0 {
            phandle = <0x00000007>;
            device_type = "cpu";
            reg = <0x00000000>;
            status = "okay";
            compatible = "riscv";
            riscv,isa = "rv64imafdch_zicsr_zifencei_zba_zbb_zbc_zbs";
            mmu-type = "riscv,sv48";
            interrupt-controller {
                #interrupt-cells = <0x00000001>;
                interrupt-controller;
                compatible = "riscv,cpu-intc";
                phandle = <0x00000008>;
            };
        };
        cpu@1 {
            phandle = <0x00000005>;
```

```
    soc {
        #address-cells = <0x00000002>;
        #size-cells = <0x00000002>;
        compatible = "simple-bus";
        ranges;
...
        uart@10000000 {
            interrupts = <0x0000000a>;
            interrupt-parent = <0x00000009>;
            clock-frequency = <0x00384000>;
            reg = <0x00000000 0x10000000 0x00000000 0x00000100>;
            compatible = "ns16550a";
        };
...
        plic@c000000 {
            phandle = <0x00000009>;
            riscv,ndev = <0x00000060>;
            reg = <0x00000000 0x0c000000 0x00000000 0x00600000>;
            interrupts-extended = <0x00000008 0x0000000b 0x0000000
            interrupt-controller;
            compatible = "sifive,plic-1.0.0", "riscv,plic0";
            #interrupt-cells = <0x00000001>;
        };
        clint@2000000 {
            interrupts-extended = <0x00000008 0x00000003 0x0000000
            reg = <0x00000000 0x02000000 0x00000000 0x00010000>;
            compatible = "sifive,clint0", "riscv,clint0";
        };
    };
};
```

devicetree
.org

Device discovery via bus controllers...

```
Gazofonias /tmp # lspci
00:00.0 Host bridge: Advanced Micro Devices, Inc. [AMD] Starship/Matisse Root Complex
00:00.2 IOMMU: Advanced Micro Devices, Inc. [AMD] Starship/Matisse IOMMU
00:01.0 Host bridge: Advanced Micro Devices, Inc. [AMD] Starship/Matisse PCIe Dummy Host Bridge
00:01.1 PCI bridge: Advanced Micro Devices, Inc. [AMD] Starship/Matisse GPP Bridge
00:01.3 PCI bridge: Advanced Micro Devices, Inc. [AMD] Starship/Matisse GPP Bridge
00:02.0 Host bridge: Advanced Micro Devices, Inc. [AMD] Starship/Matisse PCIe Dummy Host Bridge
00:03.0 Host bridge: Advanced Micro Devices, Inc. [AMD] Starship/Matisse PCIe Dummy Host Bridge
00:03.1 PCI bridge: Advanced Micro Devices, Inc. [AMD] Starship/Matisse GPP Bridge
00:04.0 Host bridge: Advanced Micro Devices, Inc. [AMD] Starship/Matisse PCIe Dummy Host Bridge
00:05.0 Host bridge: Advanced Micro Devices, Inc. [AMD] Starship/Matisse PCIe Dummy Host Bridge
00:07.0 Host bridge: Advanced Micro Devices, Inc. [AMD] Starship/Matisse PCIe Dummy Host Bridge
00:07.1 PCI bridge: Advanced Micro Devices, Inc. [AMD] Starship/Matisse Internal PCIe GPP Bridge 0 to bus[E:B]
00:08.0 Host bridge: Advanced Micro Devices, Inc. [AMD] Starship/Matisse PCIe Dummy Host Bridge
00:08.1 PCI bridge: Advanced Micro Devices, Inc. [AMD] Starship/Matisse Internal PCIe GPP Bridge 0 to bus[E:B]
00:14.0 SMBus: Advanced Micro Devices, Inc. [AMD] FCH SMBus Controller (rev 61)
00:14.3 ISA bridge: Advanced Micro Devices, Inc. [AMD] FCH LPC Bridge (rev 51)
00:18.0 Host bridge: Advanced Micro Devices, Inc. [AMD] Matisse/Vermeer Data Fabric: Device 18h; Function 0
00:18.1 Host bridge: Advanced Micro Devices, Inc. [AMD] Matisse/Vermeer Data Fabric: Device 18h; Function 1
00:18.2 Host bridge: Advanced Micro Devices, Inc. [AMD] Matisse/Vermeer Data Fabric: Device 18h; Function 2
00:18.3 Host bridge: Advanced Micro Devices, Inc. [AMD] Matisse/Vermeer Data Fabric: Device 18h; Function 3
00:18.4 Host bridge: Advanced Micro Devices, Inc. [AMD] Matisse/Vermeer Data Fabric: Device 18h; Function 4
00:18.5 Host bridge: Advanced Micro Devices, Inc. [AMD] Matisse/Vermeer Data Fabric: Device 18h; Function 5
00:18.6 Host bridge: Advanced Micro Devices, Inc. [AMD] Matisse/Vermeer Data Fabric: Device 18h; Function 6
00:18.7 Host bridge: Advanced Micro Devices, Inc. [AMD] Matisse/Vermeer Data Fabric: Device 18h; Function 7
01:00.0 Non-Volatile memory controller: Samsung Electronics Co Ltd NVMe SSD Controller 980
02:00.0 USB controller: Advanced Micro Devices, Inc. [AMD] X370 Series Chipset USB 3.1 xHCI Controller (rev 02)
02:00.1 SATA controller: Advanced Micro Devices, Inc. [AMD] X370 Series Chipset SATA Controller (rev 02)
02:00.2 PCI bridge: Advanced Micro Devices, Inc. [AMD] X370 Series Chipset PCIe Upstream Port (rev 02)
03:00.0 PCI bridge: Advanced Micro Devices, Inc. [AMD] 300 Series Chipset PCIe Port (rev 02)
03:02.0 PCI bridge: Advanced Micro Devices, Inc. [AMD] 300 Series Chipset PCIe Port (rev 02)
03:03.0 PCI bridge: Advanced Micro Devices, Inc. [AMD] 300 Series Chipset PCIe Port (rev 02)
03:04.0 PCI bridge: Advanced Micro Devices, Inc. [AMD] 300 Series Chipset PCIe Port (rev 02)
03:05.0 PCI bridge: Advanced Micro Devices, Inc. [AMD] 300 Series Chipset PCIe Port (rev 02)
03:06.0 PCI bridge: Advanced Micro Devices, Inc. [AMD] 300 Series Chipset PCIe Port (rev 02)
03:07.0 PCI bridge: Advanced Micro Devices, Inc. [AMD] 300 Series Chipset PCIe Port (rev 02)
04:00.0 USB controller: ASMedia Technology Inc. ASM1143 USB 3.1 Host Controller
05:00.0 Ethernet controller: Intel Corporation I211 Gigabit Network Connection (rev 03)
0b:00.0 PCI bridge: Advanced Micro Devices, Inc. [AMD/ATI] Navi 10 XL Upstream Port of PCI Express Switch (rev c1)
0c:00.0 PCI bridge: Advanced Micro Devices, Inc. [AMD/ATI] Navi 10 XL Downstream Port of PCI Express Switch
0d:00.0 VGA compatible controller: Advanced Micro Devices, Inc. [AMD/ATI] Navi 23 [Radeon RX 6650 XT] (rev c1)
0d:00.1 Audio device: Advanced Micro Devices, Inc. [AMD/ATI] Navi 21/23 HDMI/DP Audio Controller
0e:00.0 Non-Essential Instrumentation [1300]: Advanced Micro Devices, Inc. [AMD] Starship/Matisse PCIe Dummy Function
0f:00.0 Non-Essential Instrumentation [1300]: Advanced Micro Devices, Inc. [AMD] Starship/Matisse Reserved SPP
0f:00.1 Encryption controller: Advanced Micro Devices, Inc. [AMD] Starship/Matisse Cryptographic Coprocessor PSPCPP
0f:00.3 USB controller: Advanced Micro Devices, Inc. [AMD] Matisse USB 3.0 Host Controller
0f:00.4 Audio device: Advanced Micro Devices, Inc. [AMD] Starship/Matisse HD Audio Controller
```

```
Gazofonias /tmp # lsusb
Bus 006 Device 002: ID 0bc2:231a Seagate RSS LLC Expansion Portable
Bus 006 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 005 Device 003: ID 0d8c:0029 C-Media Electronics, Inc. the t.bone SC 360 USB
Bus 005 Device 002: ID 152d:2352 JMicron Technology Corp. / JMicron USA Technology Corp. ATA/ATAPI Bridge
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 1b1c:1c0a Corsair
Bus 001 Device 002: ID 152a:8750 Thesycon Systemsoftware & Consulting GmbH DX3 Pro
Bus 001 Device 006: ID 076b:3021 OmniKey AG CardMan 3021 / 3121
Bus 001 Device 005: ID 1b1c:1bad Corsair CORSAIR K60 RGB PRO Low Profile Mechanical Gaming Keyboard
Bus 001 Device 004: ID 1532:0084 Razer USA, Ltd RZ01-0321 Gaming Mouse [DeathAdder V2]
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

FORTH
INSTITUTE OF COMPUTER SCIENCE

The boot process

- BootROM (aka Zero-stage Boot Loader)
  - Directly addressable ROM/NOR Flash
  - Very early CPU/Platform initialization (e.g. DRAM controller, config/status registers etc)
  - Very small, very constrained
  - Unpacks/verifies (and possibly decompress/decrypt) first stage boot loader
  - No access to peripherals yet, only memory/flash
- First stage boot loader
  - Firmware and Secure monitor initialization
  - More complicated platform / SoC setup
  - Has access to storage (but usually only VFAT support)
  - May have network access (e.g. for booting over network or perform firmware update)
  - May have private storage for configuration variables
  - May even have a GUI (e.g. your PC's UEFI or older BIOS setup)
  - Fetches second stage boot loader and / or OS kernel and executes it in a less privileged mode (supervisor / hypervisor mode) using an arch-specific boot protocol or UEFI (which is arch-independent).
- Second stage boot loader (Optional)
  - Supports more options for fetching OS kernel, e.g. more filesystems, more storage devices, more OS-specific boot protocols etc

Kernel initialization overview...

- Unpack kernel to memory
- Initialize memory map
- Initialize internal states and structures (lock validator, debug objects, stack protector, cgroups etc)
- Do architecture specific initialization (CPU initialization, Detect HW layout from ACPI / Device Tree, configure hypervisor mode etc)
- Initialize the remaining kernel features, bring up the remaining cores, initialize memory management, caches, NUMA etc
- Start the High Resolution timer for the scheduler
- Initialize early console output

Kernel initialization overview...

- Save cmdline
- Starts two kernel threads, PID1 that will later become the init userspace program and PID2 that's used for adding new threads (kthreadadd)
- Begins crawling of the chain of initcalls. Initcalls are split in to arrays of function pointers aka initcall levels:
  - Core → Remaining core kernel services, platform-related (power management, buses, interrupts etc)
  - Postcore → Remaining core kernel services that depend on the above
  - Arch → Misleading name, think of it as post-postcore (same goes for most of the "levels"). It also includes some remaining arch-specific calls but it's mostly platform-related stuff (e.g. dma, clocksource etc)
  - Subsys → Kernel subsystems (e.g. pci, various buses, networking etc) and things that depend on the above
  - Fs → (pseudo) Filesystems and things that depend on the above
  - Rootfs → Mount rootfs / populate rootfs from initrd/initramfs, and things that depend on the above
  - Device → Device drivers
  - Late → Cleanup calls and things that depend on the above
  - Console → Console devices (e.g. serial)
  - Security →Linux Security Modules (SELinux, Tomoyo, Apparmor etc)
- Run the init program from rootfs, try various default paths or the one passed by the init= parameter on cmdline

PID 1...

```
Gazofonias /home/mick # lxc-start -P Containers/ -F ubuntu-test
systemd 251.4-1ubuntu7.3 running in system mode (+PAM +AUDIT +SELINUX +APPARMOR +IMA +SMACK +SECCOMP +GCRYPT -GNUTLS +OPENSSL
Detected virtualization lxc.
Detected architecture x86-64.

Welcome to Ubuntu 22.10!

Queued start job for default target Graphical Interface.
[  OK  ] Created slice Slice /system/modprobe.
[  OK  ] Created slice User and Session Slice.
[  OK  ] Started Dispatch Password Requests to Console Directory Watch.
[  OK  ] Started Forward Password Requests to Wall Directory Watch.
[  OK  ] Reached target Local Encrypted Volumes.
[  OK  ] Reached target Local Integrity Protected Volumes.
[  OK  ] Reached target Path Units.
[  OK  ] Reached target Remote File Systems.
[  OK  ] Reached target Slice Units.
[  OK  ] Reached target Swaps.
[  OK  ] Reached target Local Verity Protected Volumes.
[  OK  ] Listening on Syslog Socket.
[  OK  ] Listening on initctl Compatibility Named Pipe.
[  OK  ] Listening on Journal Socket (/dev/log).
[  OK  ] Listening on Journal Socket.
[  OK  ] Listening on Network Service Netlink Socket.
[  OK  ] Reached target Socket Units.
         Mounting POSIX Message Queue File System...
         Starting Journal Service...
         Starting Set the console keyboard layout...
         Starting Generate network units from Kernel command line...
         Starting Remount Root and Kernel File Systems...
         Starting Apply Kernel Variables...
[  OK  ] Mounted POSIX Message Queue File System.
[  OK  ] Finished Generate network units from Kernel command line.
[  OK  ] Finished Remount Root and Kernel File Systems.
[  OK  ] Reached target Preparation for Network.
         Starting Create System Users...
[  OK  ] Finished Apply Kernel Variables.
[  OK  ] Started Journal Service.
         Starting Flush Journal to Persistent Storage...
[  OK  ] Finished Create System Users.
         Starting Network Name Resolution...
[  OK  ] Reached target System Time Set.
```

```
root@ubuntu-test:~# systemctl status
● ubuntu-test
    State: running
    Units: 177 loaded (incl. loaded aliases)
     Jobs: 0 queued
   Failed: 0 units
    Since: Wed 2023-05-17 10:44:23 UTC; 46s ago
  systemd: 251.4-1ubuntu7.3
   CGroup: /
           ├─init.scope
           │ └─1 /sbin/init
           ├─system.slice
           │ ├─cron.service
           │ │ └─77 /usr/sbin/cron -f -P
           │ ├─dbus.service
           │ │ └─78 @dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
           │ ├─rsyslog.service
           │ │ └─80 /usr/sbin/rsyslogd -n -iNONE
           │ ├─systemd-journald.service
           │ │ └─41 /lib/systemd/systemd-journald
           │ ├─systemd-logind.service
           │ │ └─81 /lib/systemd/systemd-logind
           │ ├─systemd-networkd.service
           │ │ └─66 /lib/systemd/systemd-networkd
           │ └─systemd-resolved.service
           │   └─69 /lib/systemd/systemd-resolved
           └─user.slice
             └─user-0.slice
               ├─session-5.scope
               │ ├─ 84 /bin/login -p --
               │ ├─112 -bash
               │ ├─125 systemctl status
               │ └─126 pager
               └─user@0.service
                 └─init.scope
                   ├─105 /lib/systemd/systemd --user
                   └─106 "(sd-pam)"
root@ubuntu-test:~#
```

FORTH
INSTITUTE OF COMPUTER SCIENCE

Resources...

https://www.kernel.org/doc/html/latest/

https://lwn.net/

https://kernelnewbies.org/

https://elixir.bootlin.com/linux/latest/source

https://github.com/0xAX/linux-insides

https://linux-kernel-labs.github.io/refs/heads/master/lectures/intro.html

Some (outdated) books:

- Linux Device Drivers, 3rd edition
- Understanding the Linux kernel, 3rd edition
- Linux Kernel Development, 3rd edition
- Linux System Programming 2ed: Talking Directly to the Kernel and C Library

RISC-V Specific

RISC-V Privilege modes

Machine Mode

- Mandatory
- The most privileged / protected mode visible to the software (there is also Debug mode but it's only accessible / visible to hw debuggers)
- Physical memory addressing
- Physical memory protection
- Trap/Interrupt handling and delegation

User Mode

- Optional (depends on M-mode)
- The least privileged / protected mode
- Physical/virtual memory addressing Physical/virtual memory protection
- No trap/interrupt handling

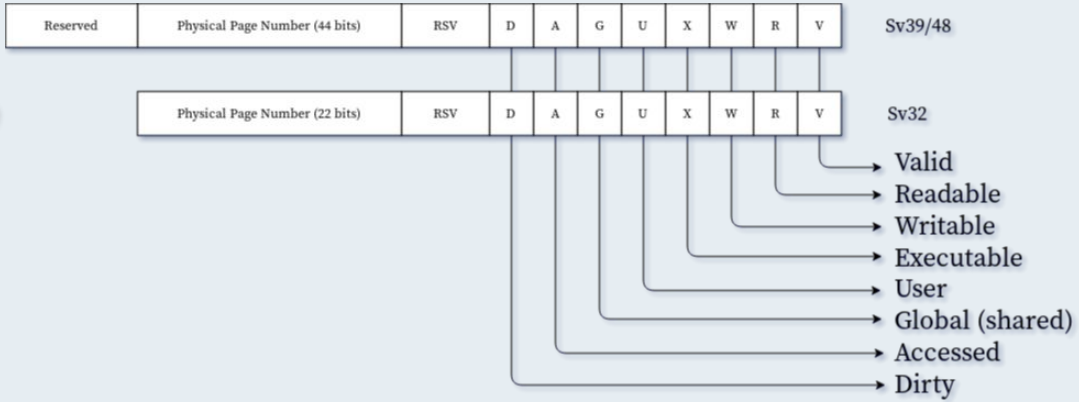
Supervisor Mode

- Optional (depends on M-mode and U-mode)
- Sits between M-mode and U-mode
- Provides virtual memory addressing / protection
- Trap/interrupt handling through delegation, managed by M-mode
- May act as a hypervisor (aka HS-mode) through the use of an extra set of CSRs, also providing a second stage of translation / protection for guests (aka VS-mode instances)

The RISC-V Privileged Spec
https://github.com/riscv/riscv-isa-manual/releases

RISC-V Virtual memory

- 3 level page table for RV32 (Sv32)
- 3, 4, 5 level page tables for RV64 (Sv39/48/57)
- 2nd stage of translation for VS mode (G-stage), managed by HS mode
- NAPOT encoding available
- Up to 16bit ASID
- SMEP always active
- SMAP controlled by sstatus.SUM bit
- Page-based memory types (Non Cacheable, I/O) for RV64



Facilities on M-mode

Provide infos on the current hardware thread (hart):

- Vendor id (mvendorid), Microarchitecture id (marchid), Implementation id (mimpid)
- Current hart's id (hartid)
- Available hart extensions (misa, menvcfg)
- Pointer to the configuration structure (mconfigptr) from which we can also generate the device tree or ACPI tables

Configure hart extensions (misa, menvcfg, mstatus) and security features (mseccfg)

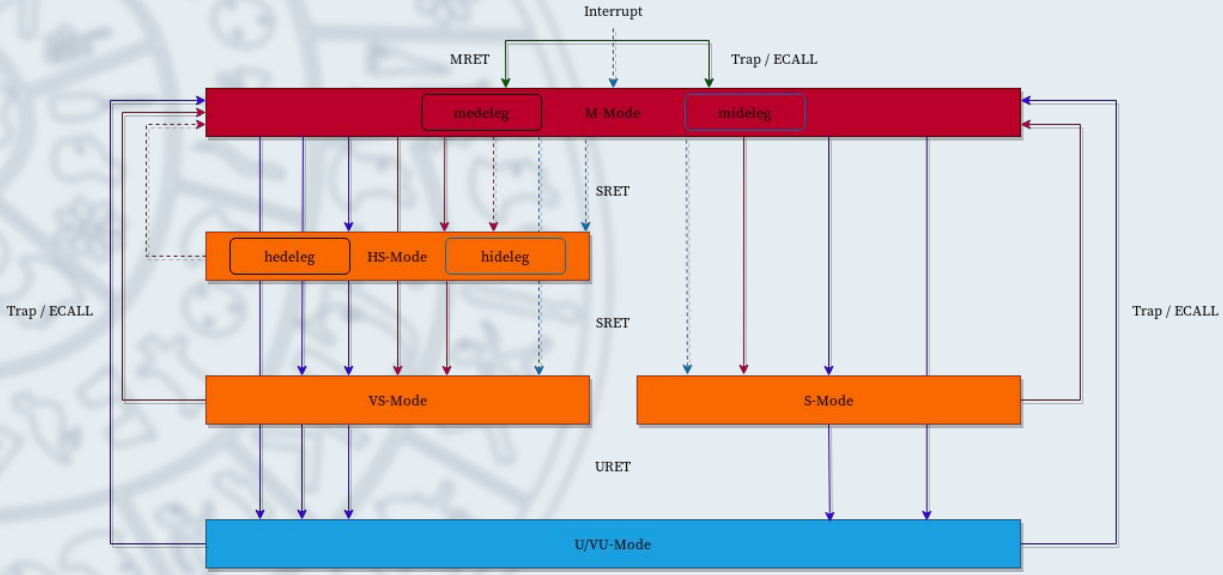
Physical Memory Protection (PMP/ePMP)

Configure profile counters

Fixed-frequency timer (mtime) with the ability to schedule timer interrupts (mtimecmp)

Configure trap and interrupt auto-delegation to S / HS modes

RISC-V Trap and interrupt delegation / mode switching



Interrupt	Exception Code	Description
1	0	*Reserved*
1	1	Supervisor software interrupt
1	2	Virtual supervisor software interrupt
1	3	Machine software interrupt
1	4	*Reserved*
1	5	Supervisor timer interrupt
1	6	Virtual supervisor timer interrupt
1	7	Machine timer interrupt
1	8	*Reserved*
1	9	Supervisor external interrupt
1	10	Virtual supervisor external interrupt
1	11	Machine external interrupt
1	12	Supervisor guest external interrupt
1	13–15	*Reserved*
1	≥16	*Designated for platform or custom use*
0	0	Instruction address misaligned
0	1	Instruction access fault
0	2	Illegal instruction
0	3	Breakpoint
0	4	Load address misaligned
0	5	Load access fault
0	6	Store/AMO address misaligned
0	7	Store/AMO access fault
0	8	Environment call from U-mode or VU-mode
0	9	Environment call from HS-mode
0	10	Environment call from VS-mode
0	11	Environment call from M-mode
0	12	Instruction page fault
0	13	Load page fault
0	14	*Reserved*
0	15	Store/AMO page fault
0	16–19	*Reserved*
0	20	Instruction guest-page fault
0	21	Load guest-page fault
0	22	Virtual instruction
0	23	Store/AMO guest-page fault
0	24–31	*Designated for custom use*
0	32–47	*Reserved*
0	48–63	*Designated for custom use*
0	≥64	*Reserved*

RISC-V Interrupt delivery

The old way (SiFive CLINT / PLIC)

- Wired interrupts only, no MSIs
- Shared between privilege modes
- Directly to M-mode and then delegated (so even S-mode software interrupts go through M-mode)
- No virtualization support

The new way (RISC-V ACLINT / AIA)

- Both wired and MSIs
- Different interrupt settings per privilege mode
- Interrupts delivered to specific privilege modes
- Virtualization support

For more information:
https://github.com/riscv/riscv-aia
https://github.com/riscv/riscv-aclint
Advanced Interrupt Architecture and Advanced CLINT
Anup Patel, John Hauser - RISC-V Summit 2021
(https://www.youtube.com/watch?v=je9Qr23mclU)

Firmware architecture

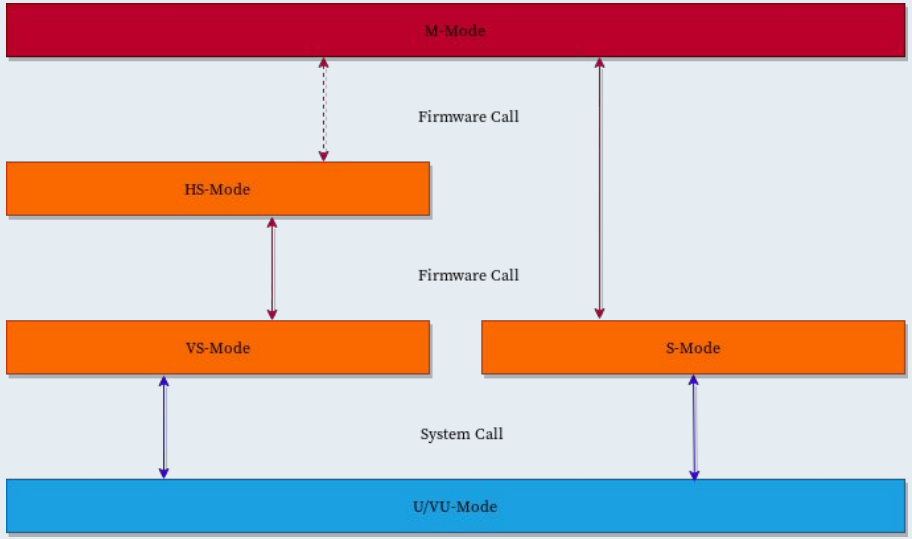
Supervisor Binary Interface (SBI)

Firmware call API

- S-Mode <-> M-Mode
- HS-Mode <-> M-Mode
- VS-Mode <-> HS-Mode

Available services:

- Provide access to M-mode facilities
  - Timer, PMU, hart/imp/vendor IDs…
- Inter-Processor Interrupts (IPI)
- Remote Fence (memory barrier)
- Hart State Management (suspend/resume)
- System Reset
- …



For more information: https://github.com/riscv-non-isa/riscv-sbi-doc

RISC-V OS Boot protocol

Direct:

- Get Device Tree through a1 gp register
- Get Hart ID through a0 gp register

RISC-V Device Tree bindings under Documentation/bindings:
/riscv/cpus.yaml
/interrupt-controller/riscv,cpu-intc.txt
/interrupt-controller/sifive,plic-1.0.0.yaml
/cpu/cpu-topology.txt

For more information:
Atish Pattra - An introduction to RISC-V Boot flow (RISC-V Summit 2019)
https://www.youtube.com/watch?v=sPjtvqfGjnY
https://archive.fosdem.org/2021/schedule/event/firmware_uor/
https://github.com/riscv-admin/riscv-uefi-edk2-docs
arch/riscv/kernel/head.S
drivers/firmware/efi/libstub/riscv-stub.c

EFI stub:

- Get Device Tree through EFI Config Table
- Get Hart ID through the device tree's chosen/boot-hartid or through the new RISCV_EFI_BOOT_PROTOCOL

devicetree
.org

FORTH
INSTITUTE OF COMPUTER SCIENCE

Runtime firmware implementations

Reference implementation: OpenSBI
Can act as a standalone firmware / first stage boot loader
Can be used as a library for other runtime firmware implementations
    Used on EDK2 (EFI Runtime firmware)
Can be used for static partitioning of the system (OpenSBI Domains)

Other implementations of the SBI spec

- Hypervisors (to provide SBI for their guests):
    - KVM, Xvisor, Diosix
- RustSBI
- Coffer (Secure monitor)

Useful links:
https://github.com/riscv-software-src/opensbi
https://github.com/xvisor/xvisor
https://diosix.org/
https://github.com/rustsbi/rustsbi
https://github.com/jwnhy/coffer

Questions ?

FORTH


Thank you !
FORTH