NØNOS Kernel Architecture

x86_64 Platform Abstraction Layer • ZeroState Privacy OS

NØNOS Kernel Subsystems

process/
Process Control Block
Context Switching
Uses: GDT, IDT, Syscall

memory/
Page Tables (4-Level)
Physical/Virtual Alloc
Uses: CPU, ACPI

sched/
Task Scheduling
Timer Interrupts
Uses: Time, IDT, APIC

interrupts/
IRQ Management
Handler Dispatch
Uses: IDT, APIC

syscall/
System Call Interface
User/Kernel Transition
Uses: arch::syscall

drivers/
Device Drivers
Hardware Abstraction
Uses: PCI, Port, DMA

fs/
Filesystem (VFS)
RAM-only Mode
Uses: Memory, Storage

security/
Capabilities
Audit, Hardening
ZeroState Security

crypto/
Ed25519, BLAKE3
ML-KEM, ML-DSA
Post-Quantum Ready

smp/
Multi-Processor
AP Startup, IPI
Uses: CPU, APIC, GDT

modules/
Signed Capsules
Sandbox, Registry
ZeroState Modules

ipc/
Inter-Process Comm
Shared Memory, Signals
Uses: Memory, Syscall

net/
Network Stack (smoltcp)
TCP/IP, Sockets
Uses: PCI, IRQ, DMA

ui/
User Interface
TUI, Graphics
Uses: VGA, Input

94 Modules | 15+ Subsystems | 256 IRQ Vectors | 6 IST Stacks | 256 Max CPUs | 4

Core Path Submodule Kernel Layer

arch/x86_64/

cpu/
CPUID Detection
Feature Flags
MSR Access
TSC Operations
Per-CPU Data

gdt/
Segments
TSS Management
IST Stacks
Selectors
SYSCALL Setup

idt/
256 Vectors
Exception Handlers
IRQ Handlers
IST Assignment
Handler Registry

interrupt/
Local APIC
x2APIC Support
I/O APIC
Legacy 8259A PIC
IPI / MSI-X

time/
TSC (Primary)
HPET
PIT (8254)
RTC (CMOS)
Unified API

syscall/
SYSCALL/SYSRET
MSR Config
Handler Dispatch
Security Policies
Statistics

acpi/
RSDP/XSDT
MADT (CPUs)
FADT (Power)
HPET/MCFG
NUMA Tables

I/O & Boot
serial/ (16550A UART)
vga/ (Text Mode)
keyboard/ (PS/2, USB)
pci/ (Config Space)
boot/ | multiboot/ | uefi/

Hardware Layer

CPU x86_64 | Memory | APIC/IOAPIC | PCI Bus | Timers | Serial/VGA | Keyboard | Storage | Network | ACPI/UEFI

NØNOS Boot Sequence

x86_64 Kernel Initialization Flow

PHASE 0
BOOTLOADER

1. **Load Kernel ELF**
   GRUB loads to memory
2. **Multiboot2 Parse**
   Boot info structure
3. **Enter Long Mode**
   Transition to 64-bit

PHASE 1
EARLY INIT

4. **Serial Init**
   COM1 @ 115200 baud
5. **CPU Detection**
   CPUID, SSE, AVX
6. **GDT + IDT**
   Segments, 256 vectors

PHASE 2
PLATFORM

7. **ACPI Parse**
   MADT, FADT, HPET
8. **APIC Setup**
   Local + I/O APIC
9. **Syscall MSRs**
   STAR, LSTAR

PHASE 3
SUBSYSTEMS

10. **Memory**
    Heap, page tables
11. **Scheduler**
    Process management
12. **Enable IRQs**
    Start scheduler

CPU Mode Transitions

Real Mode
16-bit, 1MB limit

→

Protected Mode
32-bit, GDT required

→

Long Mode
64-bit, paging on

→

KERNEL ACTIVE
All subsystems ready

BOOT TIME
~850ms

SYSTEM READY - NØNOS Kernel v0.8.0

CPU Subsystem

arch/x86_64/cpu/ - Detection, Features & Management

CPU MODULE STRUCTURE

cpuid.rs
CPUID instruction wrapper

features.rs
100+ CPU feature flags

msr.rs
Model-Specific Registers

tsc.rs
Time Stamp Counter

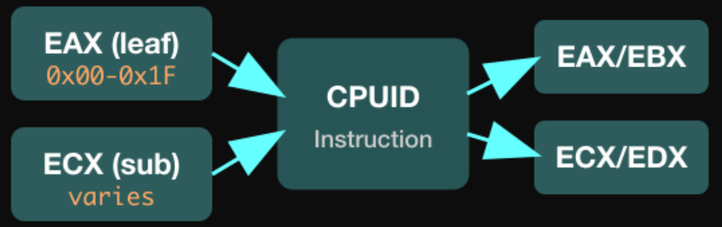
topology.rs
Core/Thread detection

per_cpu.rs
Per-CPU data (256 max)

KEY MSR REGISTERS

0xC0000080	IA32_EFER	Extended features
0xC0000081	IA32_STAR	SYSCALL segments
0xC0000082	IA32_LSTAR	SYSCALL entry
0xC0000100	IA32_FS/GS	Segment bases

CPUID INSTRUCTION FLOW



CPU TOPOLOGY

PACKAGE 0

Core 0
Thread 0, 1

Core 1
Thread 0, 1

CACHE HIERARCHY

L1
32 KB
~4 cycles

L2
256 KB
~12 cycles

L3
8-16 MB
~40 cycles

CPU FEATURES (100+ FLAGS)

SIMD
SSE, SSE2, SSE3
AVX, AVX2, AVX-512
FMA, BMI1, BMI2
+20 more

SECURITY
AES-NI, SHA
RDRAND, RDSEED
SMEP, SMAP, CET
+15 more

TIME STAMP COUNTER

```
cpu::rdtsc() // Fast read (~20 cycles)
cpu::rdtscp() // Serialized + CPU ID
```

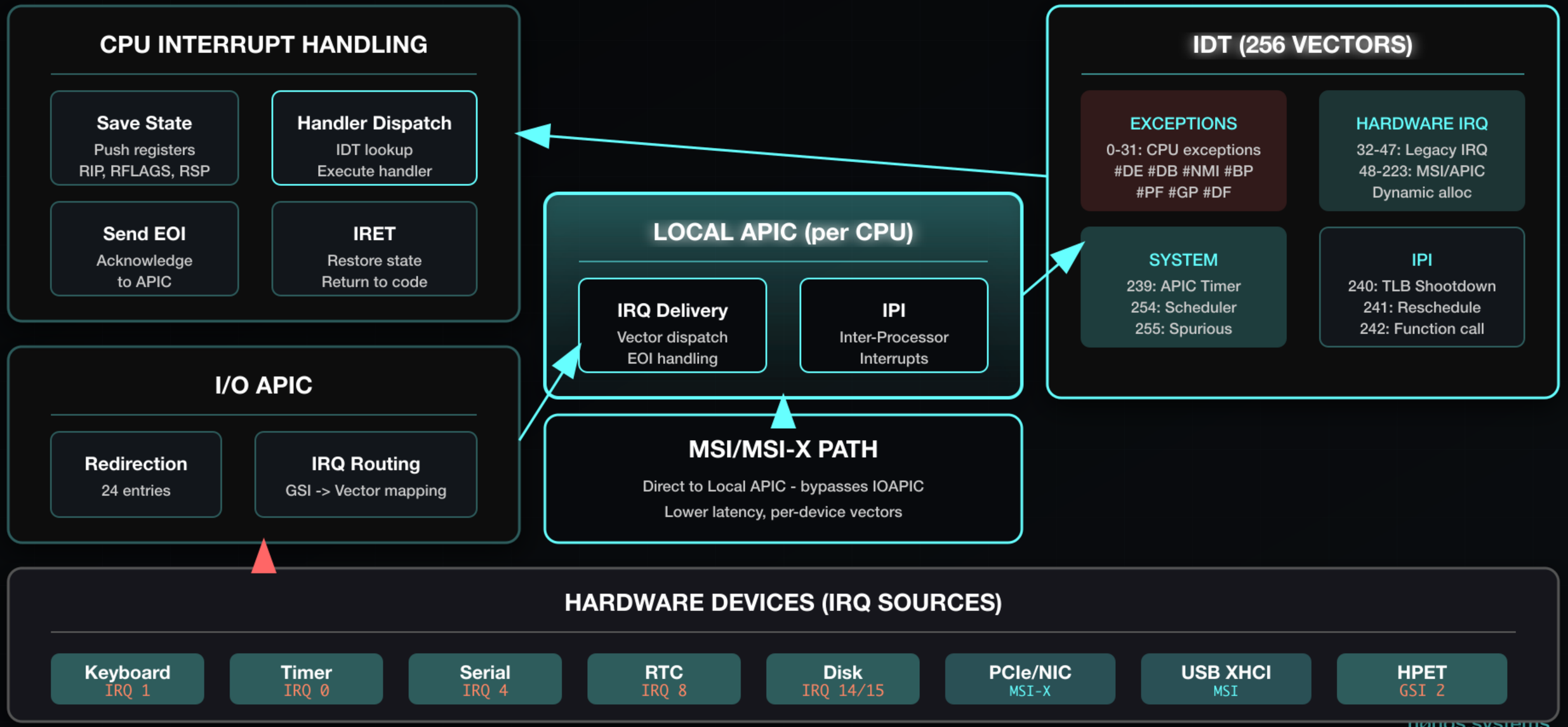
Invariant TSC: constant rate across P/C states

PER-CPU DATA

```
cpu_id: u32 // APIC ID
features: CpuFeatures // Cached
kernel_stack: u64 // via GS segment
```

Interrupt Architecture

arch/x86_64/interrupt/ - APIC, IOAPIC & IDT



Timer Subsystem

arch/x86_64/time/ - Multi-Source Timekeeping

CLOCK SOURCE HIERARCHY (Priority Order)

TSC | HPET | APIC Timer | PIT 8254 | RTC

TSC (Time Stamp Counter)

PRIMARY - Highest Resolution

```
cpu::rdtsc() // ~20 cycles
cpu::rdtscp() // Serialized + CPU ID
```

Resolution	Invariant TSC
~1 ns	Constant rate

HPET (High Precision)

SECONDARY - For Calibration

ACPI HPET table detection
Memory-mapped registers

Resolution	Frequency
~100 ns	14.3 MHz+

APIC Timer

PER-CPU - Scheduling

Local APIC timer interrupt
One-shot or periodic mode

Use Case	Mode
Preemption	TSC-deadline

UNIFIED TIME API

```
time::now() -> Instant
time::uptime() -> Duration
time::sleep(Duration)
time::delay_us(u64)
```

```
time::tsc_frequency() -> u64
time::calibrate_tsc()
time::monotonic_ns() -> u64
```

FALLBACK SOURCES

PIT 8254
Legacy timer
~840 ns
1.193 MHz

RTC (CMOS)
Wall clock time
1 second
Battery-backed

TSC Calib
PIT used to calibrate TSC frequency

SYSCALL / SYSRET Flow

arch/x86_64/syscall/ - User/Kernel Transition

USER SPACE (Ring 3)

CPL = 3

User Application

```
// libc wrapper
write(fd, buf, n) {
  syscall(SYS_write,
    fd, buf, n);
```

Register Setup

```
RAX = syscall #
RDI = arg1
RSI = arg2
RDX = arg3
R10/R8/R9 = 4-6
```

SYSCALL Instruction

KERNEL SPACE (Ring 0)

CPL = 0

SYSCALL Entry

```
RCX = user RIP (return addr)
R11 = user RFLAGS
```

Handler Dispatch

```
syscall_table[RAX](args)
// ~300 system calls
```

SYSRET

```
RAX = return value
Restore RIP, RFLAGS
```

MSR CONFIGURATION

IA32_STAR
Kernel/User CS/SS selectors

IA32_LSTAR
SYSCALL entry point (64-bit)

IA32_SFMASK
RFLAGS mask (clear IF, DF)

IA32_EFER
SCE bit enables SYSCALL

SYSCALL TABLE

File I/O read, write, open	**Process** fork, exec, exit	**Memory** mmap, brk, mprotect
Network socket, send, recv	**IPC** pipe, shmem, signal	**NØNOS** vault, crypto, zeroize

PERFORMANCE

SYSCALL/SYSRET: ~100-200 cycles | No stack switch needed | Faster than INT 0x80

Crypto Subsystem

Post-Quantum Ready Cryptographic Primitives

HASH FUNCTIONS (hash/)

- **BLAKE3** — Primary
- **SHA-512** — 512-bit
- **SHA3** — Keccak
- **SHA-256** — unified/
- **HMAC** — MAC
- **HKDF** — KDF
- **SHA-1** — Legacy only
- **Keccak256 / SHAKE128/256** — Ethereum compatible

SYMMETRIC (symmetric/)

- **ChaCha20-Poly1305** — NØNOS Primary AEAD
- **AES-256-GCM** — AEAD
- **AES-128/256** — Block cipher
- **Poly1305** — MAC

core/aead.rs - Unified AEAD traits

ASYMMETRIC (asymmetric/)

- **Ed25519** — Signatures
- **X25519** — Key exchange
- **P-256** — NIST curve
- **secp256k1** — Bitcoin/ETH
- **RSA** — Legacy
- **Curve25519** — DH base

core/traits.rs - Kem, Sig, Ed25519Sig traits

POST-QUANTUM (pqc/)

NIST PQC Standards Ready

- **ML-KEM (Kyber)** — kyber.rs - KEM
- **ML-DSA (Dilithium)** — dilithium.rs - Sig
- **SPHINCS+** — sphincs/
- **McEliece** — mceliece/
- **NTRU** — ntru/

quantum.rs - Hybrid PQ utilities

ZERO-KNOWLEDGE (zk/)

- **Groth16** — zkSNARK
- **Halo2** — Recursive
- **nonos_zk** — Attestation

zk_kernel/ - Native ZK Proofs

Pedersen | Schnorr | Sigma | Range | Equality | Membership | PLONK

syscall_zk_verify, syscall_zk_commit, syscall_zk_prove_*

APPLICATION (application/)

- **vault.rs** — Secure key storage
- **nonos_signing.rs** — Module signatures
- **ethereum/** — keccak256, ecrecover
- **certification/** — X.509 parsing

CORE API (core/)

- **api.rs** — init_crypto_subsystem generate_keypair, sign, verify
- **aead.rs** — Aead trait, aead_wrap/unwrap
- **syscall.rs** — syscall_blake3_hash, ...
- **traits.rs** — Kem, Sig, KyberKem, ...

UTILITIES (util/)

- **bigint/** — Arbitrary precision math
- **entropy.rs** — RDRAND/RDSEED
- **constant_time/** — Side-channel safe
- **rng/** — get_random_bytes

hmac.rs | misc.rs

SECURITY PROPERTIES

- **ZeroState** — Keys RAM-only, Zeroize on drop
- **Side-Channel** — Constant-time ops, No timing leaks
- **Quantum Ready** — ML-KEM + ML-DSA, Hybrid mode

Memory safety via Rust | No unsafe crypto | NIST compliance | Test vectors validated

NØNOS DEFAULT ALGORITHM SELECTION

- **Symmetric AEAD** — ChaCha20-Poly1305, Primary for all encryption
- **Signatures** — Ed25519 + ML-DSA (hybrid), Module signing, auth
- **Key Exchange** — X25519 + ML-KEM (hybrid), Forward secrecy
- **Hash** — BLAKE3 (fast) / SHA-512, Integrity, KDF input

Feature flags: mlkem512/768/1024, mldsa2/3/5, zk-halo2, zk-groth16

Driver Subsystem

drivers/ - 188 files, 23 modules - Hardware Abstract

Common Pattern

```
static CONTROLLER: spin::Once<T> = Once::new();
Single initialization, lock-free access after init
```

DRIVER PUBLIC API (mod.rs exports)

Initialization
- init_all_drivers() init_nvme()
- init_ahci() init_xhci() init_tpm()

Controller Access
- get_nvme_controller()
- get_ahci_controller() get_xhci_*

Network
- wifi_connect() wifi_scan()
- get_virtio_net_device()

TPM / Security
- measure_module() read_pcr()
- tpm_get_random_bytes()

Display / Audio
- init_gpu() with_gpu_driver()
- get_audio_controller()

Stats / Info
- get_hardware_stats()
- get_all_devices() DeviceInfo

STORAGE DRIVERS

NVMe
- NvmeController
- Admin/IO queues
- PRP/SGL lists
- Namespace mgmt

AHCI
- AhciController
- FIS-based cmds
- Command slots
- Secure erase

Controller Submodules
- controller/core.rs init.rs io.rs
- controller/admin.rs commands.rs
- dma.rs queue.rs security.rs stats.rs
- types.rs constants.rs error.rs

NETWORK DRIVERS

VirtIO-Net
- VirtQueue
- modern_regs
- smoltcp
- PacketBuf

E1000
- RX/TX rings
- EEPROM
- PHY ctrl
- checksum

Intel WiFi
- firmware
- scan/conn
- tx/rx DMA
- PCIe regs

Network Stack Integration
- VirtioNetInterface VirtioSmolBridge
- interface.rs validation.rs rate_limiter
- RTL8139 (legacy) network/stack.rs
- EtherType parsing MAC address mgmt

USB / AUDIO / DISPLAY

xHCI
- TRB rings
- Event/Cmd
- DeviceCtx
- USB 3.x

HD Audio
- BDL DMA
- Codec verb
- Streams
- PCM fmt

GPU
- VBE modes
- Framebuf
- Backbuf
- 32bpp

Controller Details
- CommandRing EventRing TransferRing
- SlotContext EpContext InputContext
- VGA text Console framebuf Keyboard
- AudioFormat StreamState DisplayMode

SECURITY / TPM

TPM 2.0
- extend_pcr_sha256()
- measure_component()
- verify_boot_chain()
- PCR 0-23 banks

security/
- validate_dma_buffer()
- validate_mmio_region()
- validate_lba_range()
- RateLimiter

Security Primitives
- safe_mmio_read32/write32
- is_config_write_allowed()
- DMA bounds LBA partition PRP validation
- Boot measurement log Quote generation

PCI/PCIe BUS LAYER

PciManager
- scan_and_collect()
- find_device_by_class()
- find_device_by_id()

ConfigSpace
- read32/write32
- BAR decoding
- 0xCF8/0xCFC ports

MSI / MSI-X
- configure_msi()
- configure_msix()
- Vector allocation

Capabilities
- enumerate_capabilities()
- get_pcie_info()
- PM, AER, SR-IOV

SecurityPolicy
- validate_config_write()
- approve_bus_master()
- Allow/Block lists

DmaEngine
- DmaDescriptor
- scatter-gather
- 64-bit addressing

PciStats
- config reads
- config writes
- violations

HARDWARE LAYER

NVMe SSD	SATA HDD	Intel NIC	VirtIO NIC	Intel WiFi	USB 3.x	HD Audio	GPU	TPM 2.0	PS/2	RTL8139
PCIe 4.0 x4	AHCI 6Gbps	E1000/E1000E	QEMU/KVM	AX200/AX210	xHCI	Intel HDA	Bochs VBE	MMIO/LPC	8042 KBC	Legacy NIC

Memory Subsystem

memory/ - 21 modules - Physical, Virtual, DMA, Security

VIRTUAL ADDRESS SPACE

USER SPACE (Ring 3)

0x0000_0000_0000_0000
0x0000_7FFF_FFFF_FFFF
128 TiB canonical low

Non-Canonical Hole

KERNEL SPACE (Ring 0)

KTEXT / KDATA
0xFFFF_FFFF_8000_0000
32+32 MiB code/data

DIRECTMAP
0xFFFF_FFFF_B000_0000
256 MiB phys map

KHEAP
0xFFFF_FF00_0000_0000
256 MiB heap

KVM (Kernel VM)
0xFFFF_FF10_0000_0000
512 MiB alloc

MMIO
0xFFFF_FF30_0000_0000
512 MiB device I/O

VMAP
0xFFFF_FF50_0000_0000
256 MiB vmalloc

DMA
0xFFFF_FF60_0000_0000
256 MiB DMA buf

FIXMAP
0xFFFF_FFA0_*
64 GiB

PERCPU
0xFFFF_FFC0_*
16 MiB/CPU

KPTI TRAMPOLINE
0xFFFF_FFFF_FFFE_0000
User/Kernel switch

4-LEVEL PAGING

PML4 (512 ent) → PDPT (512 ent) → PD (512 ent) → PT (512)

Page Sizes
4 KiB (PT) 2 MiB (PD) 1 GiB (PDPT)
PAGE_SIZE=4096 HUGE_PAGE_2M/1G

CR3
PML4 phys addr

Self-Ref
Slot 510

VIRTUAL MEMORY

virt/
map_page()
unmap_page()
translate_addr()
VmFlags

virtual_memory/
AddressSpace
VmArea
VmProtection
VmType

paging/
PageMapping
PagePermissions
tlb.rs
flush_tlb()

mmu/
init_mmu()
current_cr3()
invalidate_page()
ProtectionFlags

SECURITY / HARDENING

secure_memory/
SecurityLevel
RegionType
allocate_memory()
Capsules

safety/
validate_access()
verify_stack_integrity()
Corruption detect

hardening/
Guard pages
Stack canaries
W^X enforce
verify_kernel_*

kaslr/
validate()
verify_slide_integrity()
Randomization

UNIFIED MEMORY API

MemoryType / MemoryProtection
Anonymous | KernelCode | KernelData
UserCode | UserData | UserHeap | UserStack
Device | SecureCapsule | Shared

Core Functions
init_unified_vm()
map_memory(va, size, prot, type)
unmap_memory(va, size)
translate_virtual(va) -> PhysAddr

Address Conversion
phys_to_virt(phys) -> VirtAddr
virt_to_phys(virt) -> Option<PhysAddr>

Security
allocate_secure_region(size, owner, level)
validate_access(pid, va, write)

TLB / Page Fault
flush_tlb_range(start, size)
handle_unified_page_fault(addr, code)

PHYSICAL MEMORY

phys/
allocator.rs
bitmap.rs
Frame alloc
Zone stats

frame_alloc/
PhysFrame
AllocFlags
allocate_frame()
deallocate()

buddy_alloc/
BuddyBlock
AllocatedBlock
Power-of-2
Coalescing

page_allocator/
alloc_pages()
free_pages()
Contiguous allocation

DMA / MMIO / HEAP

dma/
DmaRegion
DmaPool
DmaConstraints
StreamingMapping

mmio/
mmio_r8/16/32/64
mmio_w8/16/32/64
MmioRegion
MmioFlags

heap/
SecureHeapAlloc
HeapStats
verify_heap_integrity()

layout/
validate_layout()
PAGE_SIZE=4096
HUGE_PAGE_2M
MAX_CPUS=64

ADDITIONAL MODULES

boot_memory/
Early alloc
Pre-heap init

region/
Memory regions
Range tracking

page_info/
Page metadata
Ref counting

proof/
Memory proofs
Verification

Statistics: HeapStats, AllocStats, VmStats
get_memory_system_stats() unified API

INITIALIZATION SEQUENCE

init_all_memory_subsystems():
1. layout::validate_layout() → 2. phys::init() → 3. frame_alloc::init()
4. heap::init() → 5. allocator::init() → 6. safety::init() → 7. kaslr::validate()

verify_all_memory_integrity()
heap_integrity + stack_integrity + kaslr_integrity + kernel_data + page_tables

KPTI (Spectre/Meltdown)

KPTI_TRAMPOLINE = 0xFFFF_FFFF_FFFE_0000
PCID_KERNEL = 0x0001 PCID_USER = 0x0002
Separate kernel/user page tables