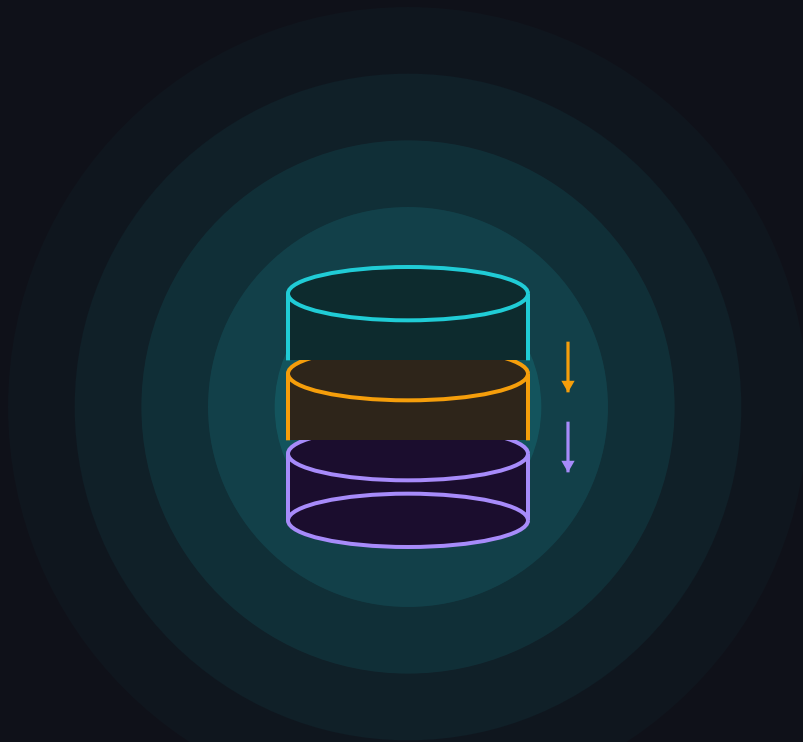




Linux Device Mapper

A Visual Guide

I/O Stack • DM Architecture • dm-linear & dm-stripe • dm-crypt & LUKS
dm-verity & Merkle Trees • Thin Provisioning • Advanced Targets • Stacking Patterns

Contents

01

Prerequisites & I/O Stack

Block devices, kernel modules, and the Linux I/O path

02

Device Mapper Architecture

Mapping tables, BIO remapping, and the DM framework

03

dm-linear & dm-stripe

Volume concatenation, LVM foundations, and RAID-0 striping

04

dm-crypt & LUKS

Transparent block encryption, key management, and LUKS headers

05

dm-verity & Merkle Trees

Integrity verification, hash trees, and Android Verified Boot

06

Thin Provisioning

On-demand allocation, snapshots, and overprovisioning

07

Advanced Targets

dm-cache, dm-integrity, dm-vdo, and more

08

Real-World Stacking

LVM on LUKS, Android boot, and composing DM layers

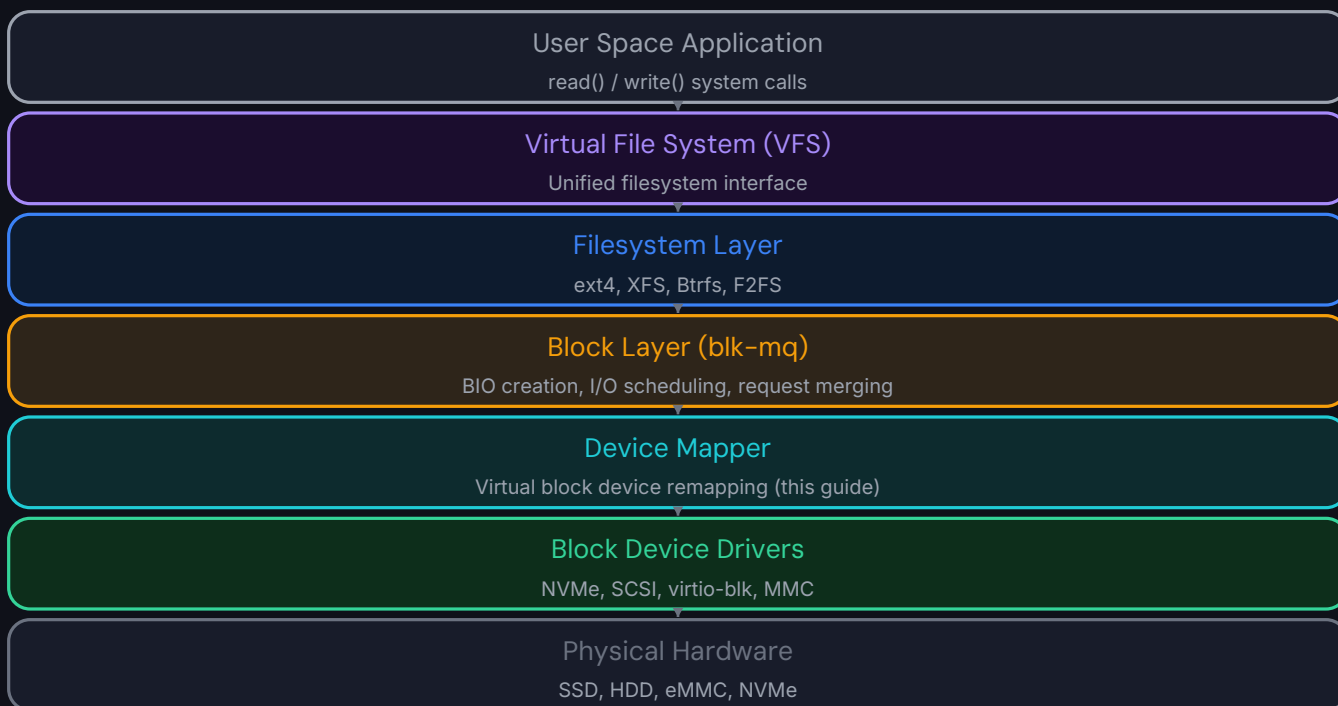
01 Prerequisites & I/O Stack

What you need to know before diving in

Before exploring Device Mapper, you should be familiar with:

- * Block devices (/dev/sda, /dev/nvme0n1) and partitions
- * Linux kernel modules and kernel vs. user space
- * Filesystem basics (ext4, XFS, read/write operations)
- * Basic cryptography (hashing, symmetric encryption)

Linux I/O Stack



Key Concept

- > Device Mapper sits between the block layer and device drivers
- > It intercepts BIOs (Block I/O) and remaps them to different devices
- > All DM operations are transparent to filesystems above