#SCaLE21x

Immutability & Atomicity in Linux

Hi, nice to meet you, I’m

Pietro di Caprio

Software Engineer @fabricators.ltd

Public Relations Manager @vanillaos.org

The problem behind mutable Operating Systems

The core issue of mutable Linux systems is their capacity for post-installation changes, compromising predictability and reproducibility. This is aggravated by a package-based structure, where independent updates or modifications to packages add layers of complexity. This further challenges the system's consistency, security, and the maintenance of a uniform state across environments.

Immutability

Once a system component is created, it remains unaltered. The software environment, once deployed, stays consistent; any change calls for a new, updated, version of the component itself.

Atomicity

Atomic operations are all-or-nothing, ensuring that tasks are completed in their entirety. This means if one part fails, the whole operation is rolled back, avoiding incomplete changes.

How do immutability solve the problem?

mainly:

- Ensuring a defined state of the system
- Preventing changes to critical system components

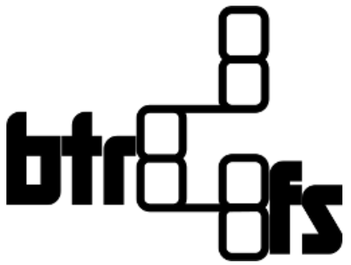
What’s the role of atomicity?

Atomicity plays a crucial role in applying system updates to immutable systems. Since immutable systems cannot be directly modified, but only updated, atomicity ensures that they are enforced as a single, indivisible unit, preventing the system from remaining in an unknown and inconsistent state.

Implementations

[A B] Root

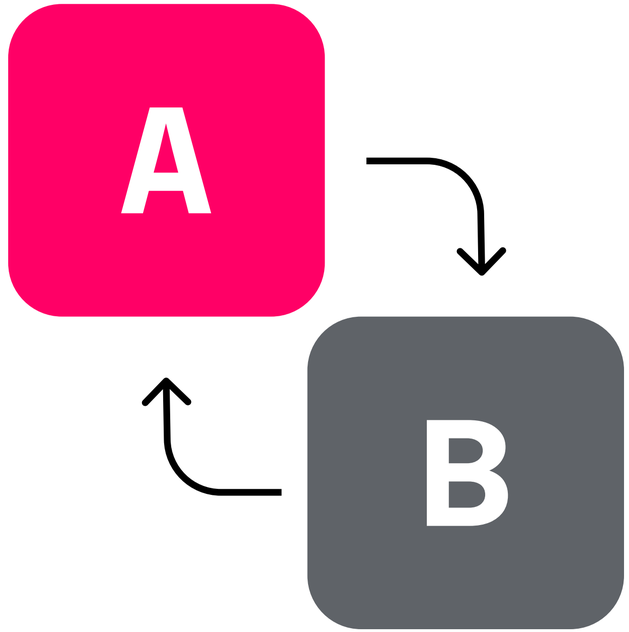
OSTree



based

[A B] Root

ABRoot brings a distro-agnostic approach to immutability, using dual partitions (A/B) to swap and apply system updates atomically. This method, employed by Vanilla OS, mirrors the update strategies of Android and Chrome OS.

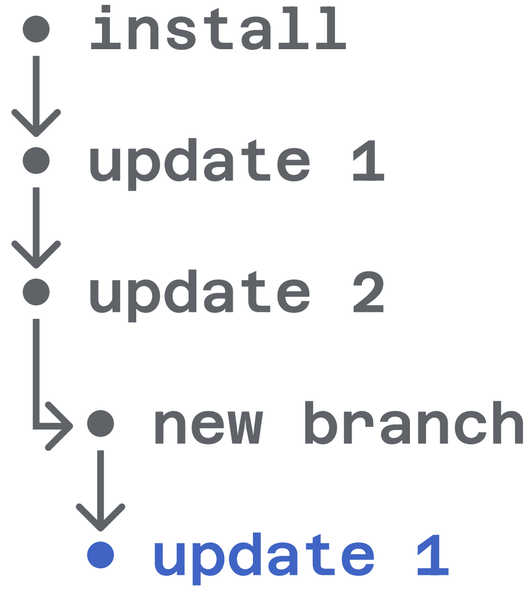


EXAMPLES

Vanilla

OSTree

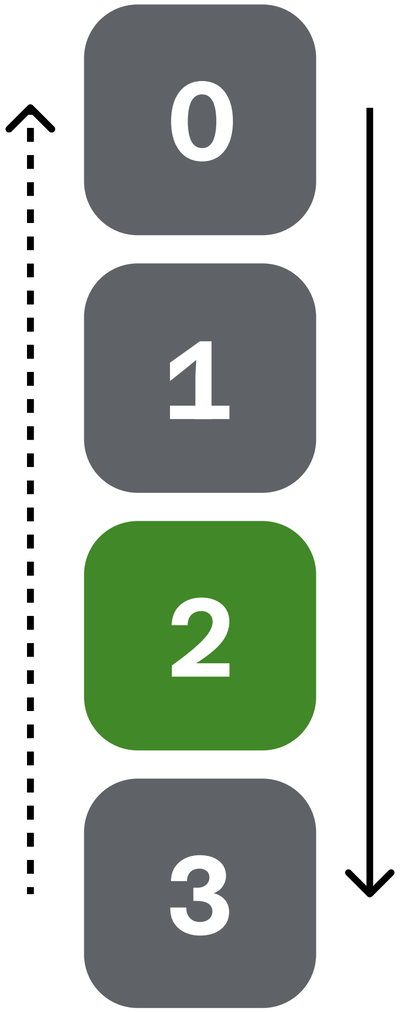
OSTree manages the system in a version-controlled manner, much like Git. It allows you to roll back to previous states easily and creates new branches for testing without affecting the main system. Some of the projects using it are: Silverblue, Endless OS and Ublue.



EXAMPLES

btrfs based

Projects like Micro OS and astOS utilize Btrfs for its immutability and atomicity features. This filesystem supports snapshots and rollbacks, enabling transactional updates where changes are applied in a single step, ensuring system integrity.



EXAMPLES

astOS

Thanks!
Any question?