LateralusOS v1.2 Architecture

SMP, capability-enforced syscalls, graphical installer

Lateralus Language

bad-antics · April 2026 · Lateralus Language Research

ABSTRACT LateralusOS v1.2 is the first release of the operating system to present a complete capability-enforced system call interface, a symmetric multiprocessor scheduler, and a graphical installer suitable for non-technical users. This report describes the kernel-level changes that made those features possible, benchmarks the SMP scheduler against the v1.1 single-processor baseline, and documents the on-disk format and recovery guarantees of the new installer.

1. Introduction

v1.1 established a monolithic but modular kernel written entirely in Lateralus, with a working VFS, process model, and framebuffer GUI. v1.2 extends that kernel in three structural directions: (i) it scales across cores, (ii) it ties every privileged operation to a capability inherited from userspace, and (iii) it ships a graphical installer that can partition, format, and bootstrap the system without dropping to a shell.

2. SMP Scheduler

The v1.1 scheduler was a single run-queue round-robin scheduler; v1.2 introduces a per-CPU run queue with work stealing, modeled on the Go runtime's M:N scheduler but specialized for kernel threads.

2.1 Design

```
struct CpuRq {
    ready: Deque<TaskId>,       // local FIFO
    stolen_from: AtomicU32,
    idle: AtomicBool,
}
static RQS: [CpuRq; MAX_CPUS]
```

Enqueue pushes onto the current CPU's local deque. Dequeue pops from the local deque; if empty, the CPU attempts to steal half of a victim's queue chosen uniformly at random. Stealing is lock-free via a Chase-Lev deque.

2.2 Benchmarks

We benchmarked a context-switch microbenchmark (two tasks, 10M switches) and the kernel's own self-compile on a 4-core QEMU guest.

- Context-switch: v1.1 took 42.1s; v1.2 takes 10.8s (3.9x).
- Kernel self-compile: v1.1 took 4m18s; v1.2 takes 1m14s on 4 cores (3.5x).
- Scalability held linearly out to 8 cores on a 16-core host; beyond 8 cores, NUMA effects dominate and we defer further study to v1.3.

3. Capability-Enforced Syscalls

Every syscall in v1.2 takes an explicit capability handle as its first argument. The kernel validates the handle against a per-process capability table before performing the operation. Capabilities are 64-bit opaque values indexed into a kernel-resident table; they can be delegated to children and revoked by the owner.

```
sys_read(cap: CapHandle, buf: *mut u8, len: usize) -> isize;
```

```
//  EBADCAP if cap does not name a readable object
//  EPERM   if cap exists but lacks Fs.Read
```

This closes a class of confused-deputy attacks that were possible in v1.1, where any process with a path could access files it should not have named. The capability table is authoritative.

4. Graphical Installer

The installer runs as a userspace program linked against the desktop framebuffer library. It performs partitioning via parted-equivalent kernel calls, formats the target volume as ext4, and writes the bootloader via a capability handed in by the loader.

4.1 Recovery

The installer records a journal of operations to a scratch partition. If the installer crashes or power is lost, the next boot replays the journal to restore the system to a consistent state. We verified this property by randomized fault injection: 10,000 trials each with a uniformly random power-cut point; 0 produced an unbootable system.

5. Other Subsystem Changes

- Ext4 read-only driver (read/write promoted to v1.3).
- USB 2.0 host controller driver (EHCI).
- Wi-Fi stack: 802.11g open/WEP/WPA2-PSK; no WPA3 yet.
- Audio: HDA output, ALSA-compatible mixer, 48kHz default.
- Virtual terminals: 6 by default, Ctrl+Alt+F1..F6.
- Secure boot chain: Shim -> GRUB -> kernel, all signed.

6. Limitations and Roadmap

v1.2 does not yet implement process migration between cores under load; work-stealing is the only balancing mechanism. It also lacks NUMA awareness and does not support hot-plugged CPUs. Wi-Fi is limited to 2.4GHz band; 5GHz is scheduled for v1.3.

The v1.3 roadmap focuses on ext4 write support, a real TCP/IP stack (v1.2 still relies on the userspace network manager), and process migration.