Correct and Performant Device Drivers via Intralingual Design

Ramla Ijaz[1], Kevin Boos[2], and Lin Zhong[1]

[1] Yale University, [2] Theseus Systems; *ramla.ijaz@yale.edu*

YALE UNIVERSITY COMPUTER SCIENCE

Theseus OS and Intralingual Design

Theseus introduced intralingual design to maximize the role of the compiler in OS design [1].

Intralingual Design Principles

1. **Match the compiler's understanding with the actual execution environment.**
   - single address space, single privilege level
2. **Enable the compiler to check OS safety and correctness invariants by subsuming resource-specific invariants into compiler ones.**
   - sharing resources via Rust's in-built reference types
3. **Lossless interfaces to preserve language level context and relationships between types.**
   - map() interface preserves relationship between virtual pages and physical frames, so they cannot be reused

Intralingual Design of Memory Management

Four invariants enforced by the type system:

1. Mapping from virtual pages to physical frames is bijective. (**bijective mapping invariant**)
2. Memory must not be accessible beyond page bounds.
3. Memory is only unmapped once, when there are no outstanding references.
4. A memory region must only be mutable or executable if mapped as such.

Intralingual + Verification

Motivation

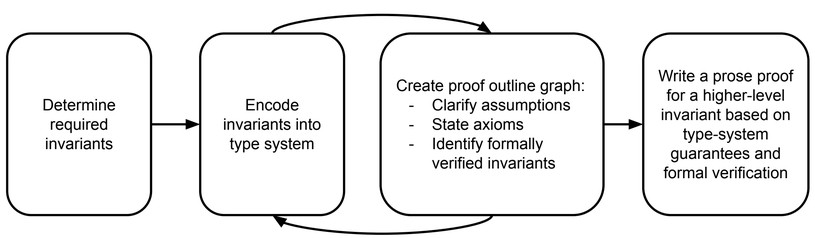
- Bug in frame allocator violated bijective mapping invariant
- Duplicate frames created → multiple pages mapped to same frame
- Kept overwriting DMA memory in a network device driver

Key Idea

Intralingual invariants assume correctness of the compiler and manually-inspected code.

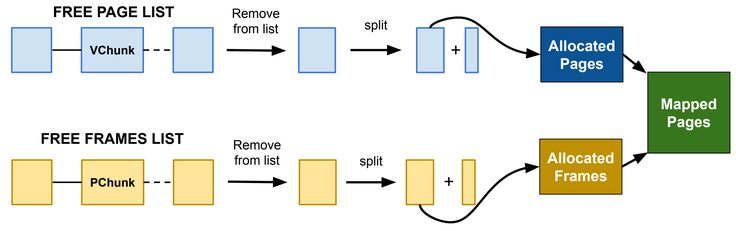
- Correctness of higher-level invariants can be traced back to correct implementation of lower-level invariants
- Modular verification of lower-level invariants
- Increase reliability of system invariants without proof burden of full system verification

Design Methodology



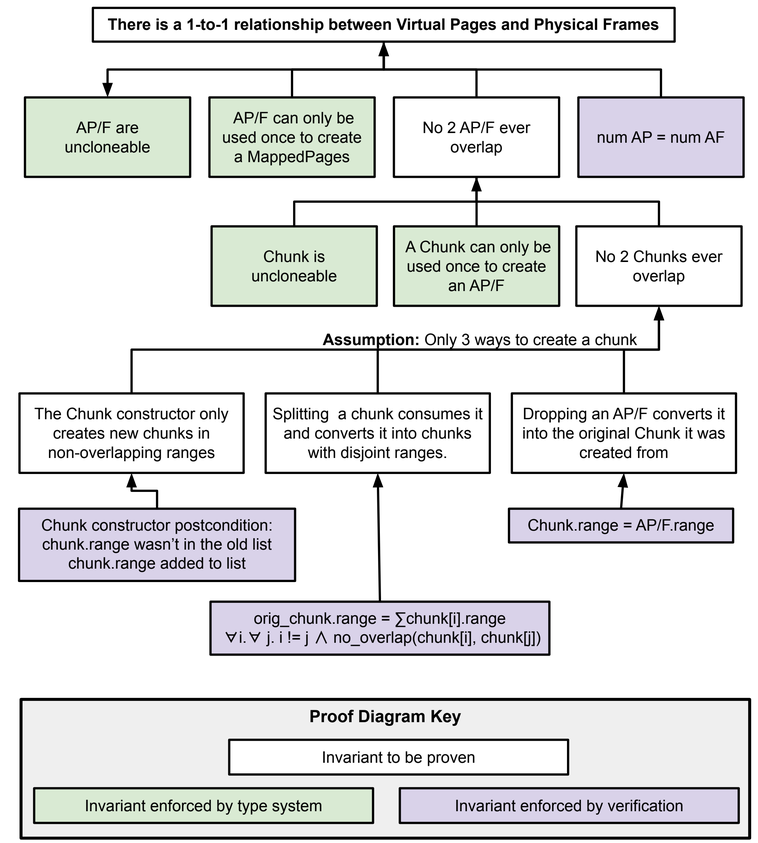
Memory Management Subsystem

Creation of `MappedPages`



`Chunk` = A range of pages/frames that are unallocated
`AllocatedPages(AP)` = A range of pages that have been allocated
`AllocatedFrames(AF)` = A range of frames that have been allocated

Proof Outline of Bijective Mapping Invariant



[1] Boos, Kevin, et al. "Theseus: an experiment in operating system structure and state management." 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20). 2020.