

# Design and Implementation of an OS for Use in the Operating Systems Lecture Track

Students



Fabian Imhof



Matteo Gmür

**Introduction:** This paper examines all aspects of memory management in a custom operating system. The goal is to create a foundation for an interactive learning tool to be used in the "Operating Systems 1" and "Operating Systems 2" lectures at OST. Topics covered include parsing Multiboot2 information, implementing frame allocation and page tables, and using these components to remap the kernel. Based on this, dynamic memory allocation during kernel runtime was implemented using different allocation strategies.

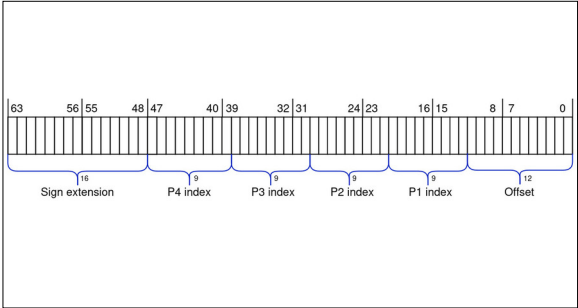
**Objective:** The objective of this work is to develop a modular and extensible memory management system encompassing physical and virtual memory management as well as heap management. This provides a robust foundation for extending other kernel components while serving as a learning platform for hands-on exploration of operating system concepts.

**Result:** The developed operating system includes:

- Physical memory allocator: Manages the allocation and deallocation of physical memory, including a simplistic version without deallocation functionality.
- Virtual memory management: Implements multilevel page tables to separate logical and physical address spaces.
- Heap memory allocator: Enables dynamic memory allocation within the kernel heap.

This foundation can be extended in the future to include additional kernel functionalities and to expand the feature set of the existing memory management system.

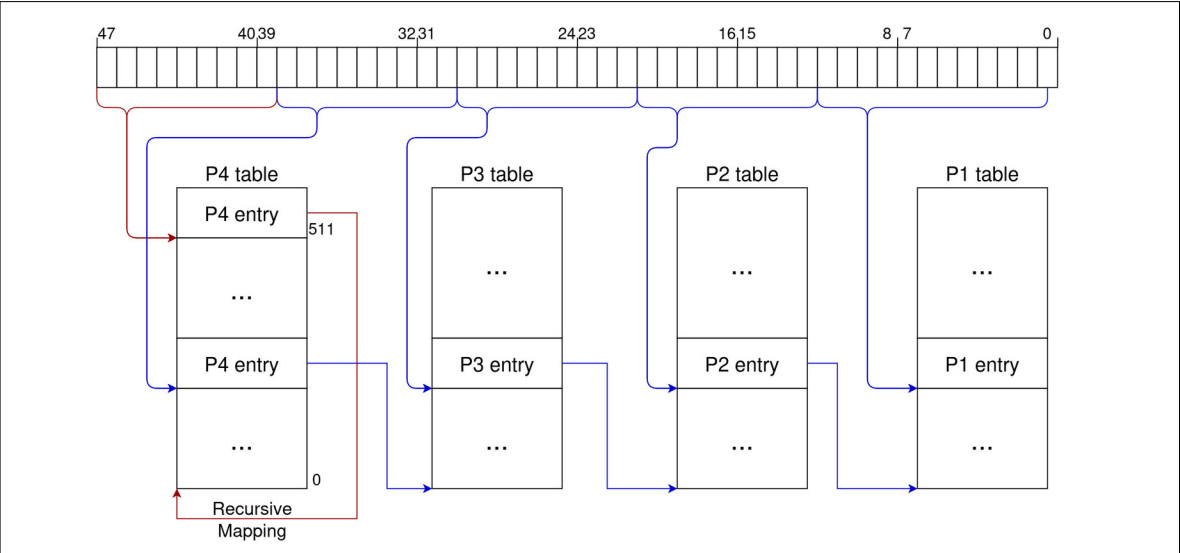
**Sketch of virtual address structure**  
Oppermann, P. (Accessed: 15–12–2024). Writing an os in rust.



**Terminal output after GRUB boot of the kernel**  
Own presentment

```
TeachOS is starting up...
Kernel remapping successful
Heap remapping successful
Heap test successful
TeachOS Panic: Architecture specific main returned!
```

**Diagram showing the last P4 entry containing the address of the P4 table start instead of an entry in the P3 table**  
Oppermann, P. (Accessed: 15–12–2024). Writing an os in rust.



**Advisor**  
Felix Morgner

**Subject Area**  
Software Engineering -  
Core Systems, System  
Software