mirx

Rust Microkernel Implementation Checklist

1. Bootstrapping & Initialization

(I used ready to use bootloader just to get around of bootloader)

- [ ] Detect memory map / implement physical memory manager (frame allocator)
- [ ] Initialize early kernel environment (stack, heap setup)

2. Basic Kernel Infrastructure

- [ ] Implement minimal panic handler & logging/debug output (e.g., serial console)
- [ ] Create simple kernel heap allocator (e.g., bump allocator)
- [ ] Define basic error handling strategy

3. Memory Management

- [ ] Setup virtual memory (paging)
- [ ] Implement page table management and mapping primitives
- [ ] Enable dynamic allocation support in kernel

4. Task & Thread Management

- [ ] Define Task Control Block (TCB) data structure
- [ ] Implement context switch mechanism
- [ ] Setup timer interrupt & preemptive multitasking basics
- [ ] Create, schedule, and switch between multiple tasks

PROF

5. Inter-Process Communication (IPC)

- [ ] Design message passing primitives (channels/queues)
- [ ] Implement basic synchronous IPC between tasks
- [ ] (Optional) Setup shared memory or buffer passing

6. Hardware Abstraction

- [ ] Setup interrupt handling & vector table
- [ ] Implement basic device driver interface (e.g., timer, keyboard)
- [ ] Implement timer driver for scheduling and timekeeping

7. User Mode Support

- [ ] Define user/kernel mode privilege separation
- [ ] Setup user mode stack and page tables

- [ ] Implement basic system call interface

8. File System / Storage (Optional early)

- [ ] Design minimal virtual filesystem interface
- [ ] Implement basic block device driver

9. Networking (Optional early)

- [ ] Implement basic network device driver
- [ ] Create simple packet send/receive primitives