CSCE 613 Final Project - Extended Rust Micro Kernel

Name: Yanze Li UIN: 326006687

Project Description

This goal of this project is using Rust programming language to implement a micro-kernel prototype. The intended features includes **Frame Allocator**, **Heap Allocator**, **Thread**, and **Process**.

The starting point of the project is a minimal bootable kernel with interrupt and VGA text mode support, plus a very limited frame allocator.

Therefore, in my project, I re-implemented the frame allocator from scratch and build a kernel heap allocator based on it. For the thread/process implementation, it is only required to support single-threaded process, therefore, the key difference between the thread and process in my project whether having a separate memory space.

By the time this report was written, the frame allocator and heap allocator has been fully implemented, with some passed test cases available. The heap allocator can successfully replace the global heap allocator, so we can use `Box` and `vec!` in the rest implementation. The implementation of thread/process is only partly done. I'm working on implementing the rest, and hopefully this will be done by the review session.

References

For frame allocation and heap allocation, following materials are referenced:

Design

Implementation

Conclusion & Future Work

link text