19BCE2311 Gaurav Singh

19BCE2347 Rajvi Jasani

19BCE0291 Vinayak Dubey

Aim: To make a simple Operating System programmed in Rust programming Language for x86\_64 architecture. (Tested on QEMU virtual machine) and coded **without** using the **std library.**

The Operating System shall have:

1. A kernel able to interact with the system hardware components such as:
   1. The screen of the system (at VGA resolution {640\*480 px})
   2. System memory, using memory allocation algorithms

Eg: bump allocation and linked list allocation.

1. The kernel will able to:
   1. Find double faults (failure to invoke specific exception handler) by setting up an Interrupts stack table to catch double faults
   2. Acknowledge hardware interrupts such as keyboard interrupt, and get keyboard input onto the screen
   3. Implement paging, for better and more efficient memory allocation, using virtual memory and multilevel page tables.
   4. Implementing heap memory allocation, for dynamic memory use.

The required OS uses basic and essential principles of:

1. Memory allocation
2. Exception handling
3. Interrupts
4. Display information onto screen
5. System level security

**NOTE**: The OS will be written in Rust programming language, which relies on the concept of ownership and borrowing, thus ensuring memory safety while memory is being accessed. Making the OS secure to use.