DM510 – Operating Systems

Assignment 1: System Call

Yunus Emre Balci (yubal22)

[yubal22@student.sdu.dk](mailto:yubal22@student.sdu.dk)

Jenin Imad El-Merie (jeelm22)

[jeelm22@student.sdu.dk](mailto:jeelm22@student.sdu.dk)

Salma Rashid Jamale Khair (sahka22)

[sahka22@student.sdu.dk](mailto:sahka22@student.sdu.dk)

Southern University of Denmark

March 2nd. 2024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Contents

[1. Introduction 3](#_Toc160135090)

[2. Design decisions 3](#_Toc160135091)

[3. Implementation 4](#_Toc160135092)

# Introduction

This assignment is made in collaboration with Jenin Imad Merie (jeelm22) and Salma Rashid (sakha22).

Our goal for this assignment is to add two system calls to User-mode Linux (UML) to implement a message box in the kernel space. These system calls are designed to work with a message box located within the kernel’s memory space, allowing different processes to send and receive short messages or byte arrays to one another. This message box is implemented as a stack, meaning that messages to be stacked on top of each other as they come in. They’re also read from the top as well. It is a straightforward, simple, and yet effective way to manage interprocess communication (IPC).

# Design decisions

As mentioned in the assignment, the decision to implement the message box as a stack is because of the simplicity and efficiency. The design we use, messages are added to the top of the stack with **sys\_dm510\_msgbox\_put** system call by pushing and retrieved from the top using the **sys\_dm510\_msgbox\_get** system call by popping. By this approach, it ensures that the most recent message is always accessed first.

A design element is our concurrency management, implemented through a spinlock, **msgbox\_lock**. This ensures that when one process is being executed, **sys\_dm510\_msgbox\_put,** to push a message, no other process can interrupt or access the stack at the same time. This way, we keep everything in order and ensures that the messages do not get mixed up or lost.

# Implementation

In our implementation of the message box, it involves two key systems:

* **sys\_dm510\_msgbox\_put**, messages being pushed on to the stack.
* **sys\_dm510\_msgbox\_get**, messages being popped off.

## Pushing Messages

In the