Simple Operating System Kernel

A PROJECT REPORT

Submitted to

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING IN COMPUTER SCIENCE ENGINEERING

By SANJAY KUMAR SOY 192311151

Supervisor Dr. GURURAMA SENTHIL Pandiarajan

CSA0460- Operating Systems for Resource Allocation



SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES, CHENNAI-602 105.

December 2024

**Abstract**

This project aims to design and implement a basic operating system kernel that demonstrates fundamental operating system concepts, including process management, memory management, and input/output operations. The kernel is developed using Assembly and C programming languages and can be tested in an emulator environment.

**1. Introduction**

Operating systems are critical software that manage computer hardware and provide services for computer programs. This capstone project focuses on creating a simple operating system kernel to understand the underlying principles of operating systems.

**2. Objectives**

- To develop a basic bootloader that initializes the hardware and loads the kernel.

- To implement a simple kernel capable of displaying messages on the screen.

- To gain hands-on experience with low-level programming and operating system concepts.

**3. Components**

Bootloader

The bootloader is responsible for loading the kernel into memory when the computer starts up.

Kernel

The kernel manages processes, memory, and handles basic input/output operations.

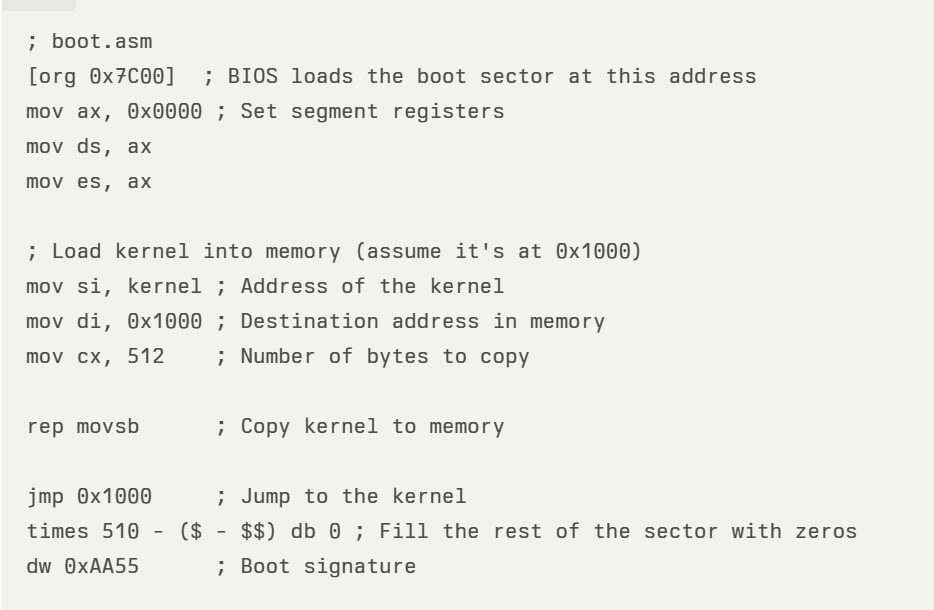
Shell

A simple command-line interface to interact with the kernel (not implemented in this version).

**4. Implementation**

Bootloader Code

The bootloader is written in Assembly language:



Kernel Code

The kernel is written in C:



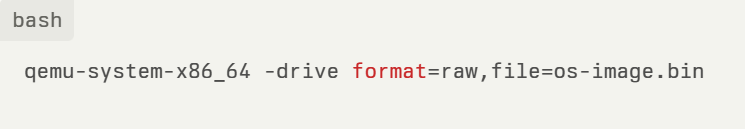
**5. Compilation and Linking**

To compile and link the bootloader and kernel, use the following commands:



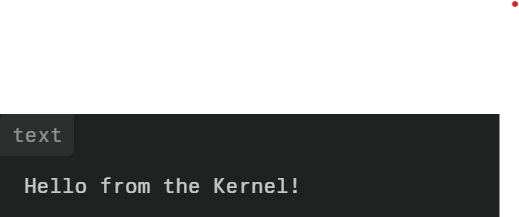
**6. Testing the Operating System**

The operating system can be tested using QEMU or another emulator. Use the following command to run your OS:



**Output**

Upon running your OS in QEMU, you should see:



This output indicates that your bootloader has successfully loaded the kernel, which has printed a message to the screen.

**7. Conclusion**

This capstone project successfully demonstrates the basic principles of operating systems through the development of a simple kernel. The project provides valuable insights into low-level programming and operating system functionality.

**8. Future Work**

Future enhancements could include:

- Implementing multi-tasking capabilities.

- Adding a file system for data storage.

- Developing a more advanced shell interface for user interaction.

- Integrating network functionalities.