Mini OS GUI Project Report

Project Overview

The Mini OS GUI is a simulated operating system interface built using Qt framework. It provides a graphical user interface that mimics the functionality of a simple operating system with multiple applications and features. The project demonstrates concepts of operating systems, resource allocation, process management, and GUI application development.

Project Structure

The project consists of several key components:

1. **Main Dashboard**: Central hub for accessing all applications and features
2. **User Authentication**: Boot sequence with user mode selection
3. **Resource Allocation**: System for allocating CPU and memory resources
4. **Applications**:
   - Text Editor
   - Calculator
   - Calendar
   - File Browser
   - Mini Games
   - System Information/PCB Tables

Key Features

Boot Sequence

- Splash screen with team credits
- Mode selection (User/Kernel)
- Resource allocation interface

Main Dashboard

The dashboard serves as the central navigation hub, providing access to all applications through a clean, intuitive interface.

Application Suite

- **Text Editor**: Create, edit, and save text files
- **Calculator**: Perform basic arithmetic operations
- **Calendar**: View and navigate dates
- **File Browser**: Navigate the file system

- **Mini Games**: Collection of simple games for entertainment
- **System Information**: View process and resource information

Resource Management
- CPU core allocation
- Memory allocation
- Process creation and termination tracking

Technology Stack

- **UI Framework**: Qt
- **Programming Language**: C++
- **Build System**: CMake
- **Testing Framework**: Custom Python-based automation tool

Testing Tool

The project includes a custom Python-based testing and automation tool (`new.py`) that allows for:
- Automated testing of different application scenarios
- Simulation of user interactions
- Logging test results
- Verifying application behavior

Implementation Details

UI Design
The interface uses a modern, clean design with:
- Consistent styling across all applications
- Intuitive navigation
- Responsive layouts

Architecture
- Modular design with separate components for each application
- Object-oriented approach with clear class hierarchies
- Event-driven programming model

Mini Games Implementation
The Mini Games module is implemented as a separate dialog with a stacked widget design, allowing users to switch between different games while maintaining a consistent interface.

Execution Instructions

Prerequisites

- Linux-based operating system (Ubuntu recommended)
- Qt libraries installed
- C++ compiler (GCC or similar)

Building the Project

1. Clone the repository:

```
Git clone  https://github.com/zaidhassan2/Operating-System.git
cd Operating-System
```

2. Build the project:

```
mkdir build
cd build
cmake
make
```

3. Run the application:

```
./myos-gui
```

Using the Application

1. **Startup**:
   - The application will display a splash screen
   - Select the desired mode (User Mode recommended)
   - Allocate resources as prompted

2. **Dashboard Navigation**:
   - Click on application icons to launch different features
   - Multiple applications can be open simultaneously

3. **Text Editor**:
   - Create new files with the "New" button
   - Save files with the "Save" button
   - Open existing files with the "Open" button

4. **File Browser**:
   - Navigate through directories
   - View files and their properties
   - Select files to open with appropriate applications

5. **Mini Games**:
   - Access through the Mini Games icon on the dashboard
   - Select from available games
   - Use the close button to return to the dashboard

Running the Testing Tool

The project includes a Python-based testing and automation tool for verifying functionality:

1. Install required Python dependencies:

```
pip install tkinter
```

2. Run the testing tool:

```
python /path/to/new.py
```

3. In the testing tool:
   - Verify the path to the Mini OS GUI executable
   - Select a test scenario from the dropdown
   - Click "Start Testing" to run the automated test
   - View logs in the execution log area
   - Save logs for future reference using the "Save Log" button

Conclusion

The Mini OS GUI project demonstrates the implementation of a simulated operating system interface with multiple applications and features. It provides a practical example of GUI

application development, resource management, and process handling in a user-friendly environment.

The modular design allows for easy extension with additional applications and features, making it a valuable learning tool for understanding operating system concepts and GUI application development.

If you found any problem, feel free to open to report it, or can contact me freely.

Good Luck!