MiniOS Project Report

Md Touhiduzzaman Saiem
ID: 2233356642
Course: CSE323
Section: 14

December 26, 2025

Contents

1 Introduction

The MiniOS project is a simulation of a basic operating system that demonstrates process management, memory management with paging, resource allocation using the Banker's Algorithm, and a simple file system. The purpose of this project is to understand key OS concepts in a controlled environment.

2 Project Overview

2.1 Features Implemented

- Process creation and management using Process Control Blocks (PCBs)
- Memory management with paging and frame allocation
- Logical to physical address translation
- Deadlock avoidance using Banker's Algorithm
- Round-Robin CPU scheduling
- Simple file storage and retrieval system
- Stress testing with multiple processes, resource requests, and file operations

2.2 Data Structures

- **PCB:** Holds PID, state, burst time, remaining time, pages, and page table entries
- **Page Table:** Each entry contains frame number and validity bit
- **File System:** Array of file structures with name, data, and usage flag
- **Resource Allocation Matrices:** Maximum need, allocated, and remaining need for each process

3 Implementation Details

3.1 Process Management

Processes are created dynamically with a specified burst time and memory size. Each process is initialized in the READY state and assigned a page table with allocated frames.

3.2 Memory Management

Paging divides memory into fixed-size frames. Logical addresses are translated to physical addresses using the page table. Page faults are detected when a page is not in memory.

3.3 Resource Allocation

The system uses Banker's Algorithm to handle multiple resource types safely. Requests are granted only if they maintain a safe state.

3.4 Scheduler

A Round-Robin scheduler with a time quantum of 2 units ensures fair CPU time distribution among READY processes.

3.5 File System

The file system allows storing and retrieving small text files. Each file has a name, data, and usage flag.

4 Challenges and Solutions

Situation

While developing MiniOS, I faced challenges in integrating process management, memory paging, resource allocation, and file operations into a stable system.

Task

I needed to simulate multiple processes with dynamic memory and safe resource allocation while providing interactive shell commands.

Action

I implemented:

- Integrated process creation, paging, and Banker's Algorithm.
- Round-Robin scheduler to manage CPU time.
- Interactive shell for creating processes, requesting resources, translating addresses, and managing files.
- Stress tests for multiple processes and random resource requests.

Result

- Successfully handled multiple processes and resources.
- Prevented deadlocks using Banker's Algorithm.
- Demonstrated proper process state transitions and memory management.
- Achieved system stability under stress conditions.

5 GitHub Repository and Video Demo

The project folder is available on GitHub:
https://github.com/TOUHID192157/Cse$_3$23$_m iniOs$
A 2–5 minute video demo showcases:

- Booting MiniOS
- Creating processes
- Requesting resources and demonstrating Banker's Algorithm
- Memory dumps and logical address translation
- File operations: storing and reading files
- Running the scheduler and completing processes

6 Analysis and Conclusion

The MiniOS stress test demonstrates:

- Correct process state transitions (READY, RUNNING, TERMINATED)
- Accurate memory management and address translation
- Safe resource allocation preventing deadlocks
- Robust file storage and retrieval
- Stable system operation under multiple processes

This project highlights my skills in C programming, process scheduling, memory handling, resource management, and system integration. MiniOS provides a practical, hands-on understanding of core operating system concepts.