Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Spring, Year: 2024), B.Sc. in CSE (Day)

LAB PROJECT PROPOSAL

Operating System Algorithm Simulator

Course Title: Operating System Lab
Course Code: CSE 310
Section: 213 D4

Students Details

Name	ID
Rabby Khan	213902037
Mostak Ahmmed	213902126

Submission Date: 09- 05 - 2024
Course Teacher's Name: Md. Shoab Alam

Lab Project Status	
Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Overview

Operating systems are vital software systems that manage computer hardware and software resources, providing fundamental services to applications and users. Understanding and simulating operating system services through practical experiments can greatly enhance comprehension of their behavior and efficiency. This project aims to simulate key operating system services—specifically CPU scheduling algorithms, page replacement strategies, and contiguous memory allocation methods—using shell scripting. In this project, we leverage the simplicity and flexibility of shell scripting to model and simulate these critical operating system functionalities. By employing shell scripts, we can create controlled environments that mimic real-world scenarios, enabling us to study and compare the performance of various algorithms under different conditions. The project focuses on fundamental aspects of operating system design and aims to provide insights into the strengths and limitations of each simulated service

This project a structured approach to simulating operating system services using shell scripting, focusing on CPU scheduling, page replacement, and memory allocation.The shell script features a user-friendly, menu-driven interface that allows users to interactively choose and execute simulated OS services. Each service is encapsulated within functions that handle specific functionalities, incorporating error checking and user prompts to enhance usability.

1.2 Motivation

The motivation behind undertaking this project lies in the desire to deepen our understanding of operating system principles and explore the capabilities of shell scripting as a practical tool for simulating core OS services. Several key factors drove the selection and execution of this project:

1.Educational Value:

Operating system services are fundamental concepts in computer science and software engineering curricula. By simulating these services, we aimed to reinforce theoretical

knowledge with hands-on experience, enhancing comprehension of OS functionalities.

2.Practical Application of Shell Scripting:

Shell scripting is a ubiquitous skill in Unix-like environments, offering a lightweight yet powerful means of automating tasks and interacting with system resources. Exploring its potential to simulate OS services presented an opportunity to leverage scripting languages in a novel context.

3.Demonstrating Fundamental OS Concepts:

Simulating process management, file handling, and resource allocation allowed us to illustrate foundational OS concepts in a controlled environment. This hands-on approach provides a tangible demonstration of how OS services operate and interact.

4.Innovation in Simulation Techniques:

Leveraging shell scripting for OS simulation offered a unique perspective on simulation techniques. The project explored innovative ways to replicate complex behaviors using lightweight scripting languages.

Overall, the motivation behind this project was rooted in the aspiration to combine theoretical knowledge with practical implementation, leveraging scripting languages to emulate critical OS services and gain valuable insights into system behaviors and interactions. The project aimed to inspire curiosity, innovation, and skill development in the realm of operating systems and scripting technologies.

1.3 Problem Definition

1.3.1 Problem Statement

The project aims to address the challenge of simulating fundamental operating system services using shell scripting within a Unix-like environment. Specifically, the goal is to develop a shell script capable of emulating key OS functionalities such as process management, file handling, and resource allocation to enhance understanding of operating system concepts and demonstrate the practical utility of scripting languages.

1.Script Complexity: Designing a shell script that effectively simulates diverse OS services while maintaining clarity and modularity in the codebase.

2.System Interaction: Ensuring seamless interaction with system utilities and commands to mimic real-world OS behaviors within the script's execution environment.

3.Error Handling: Implementing robust error handling mechanisms to manage unexpected inputs or system responses during the simulation.

4.Resource Simulation: Developing methods to emulate resource allocation and management, including memory allocation, file operations, and process scheduling.

5.Performance Considerations: Optimizing the script for efficient execution and minimal resource consumption to simulate OS services effectively.

The project seeks to address these challenges by leveraging shell scripting techniques, Unix-like system utilities, and a structured approach to simulation design and implementation. By developing a comprehensive shell script capable of simulating essential OS services, the project aims to provide a valuable educational tool and practical demonstration of scripting languages in system simulation and automation.

1.3.2 Complex Engineering Problem

The project addresses a complex engineering problem involving the simulation of fundamental operating system services using shell scripting. This endeavor presents several intricate challenges and considerations:

Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

Table 1.1: Summary of the attributes touched by the mentioned projects

Name of the P Attributess	**Explain how to address**
P1: Depth of knowledge required	Cannot be resolved without in-depth engineering knowledge at the level of one or more of K3, K4, K5, K6 or K8 which allows a fundamentals-based, first principles analytical approach
P2: Range of conflicting requirements	Involve wide-ranging or conflicting technical, engineering and other issues
P4: Familiarity of issues	Involve infrequently encountered issues

1.4 Design Goals/Objectives

The objective of this project is to simulate fundamental operating system services using shell scripting. Specifically, this project focuses on simulating CPU scheduling algorithms, page replacement algorithms, and contiguous memory allocation algorithms.

The project is divided into three main components, each simulating a different operating system service:

1.CPU Scheduling Algorithms: Simulate and compare various CPU scheduling algorithms, such as First Come, First Serve (FCFS), Shortest Job First (SJF), Round Robin (RR). under different process workloads.

2.Page Replacement Algorithms: Implement and analyze page replacement algorithms, including First In, First Out (FIFO), Least Recently Used (LRU), and Optimal Page Replacement, to understand their impact on virtual memory management and page fault rates.

3.Contiguous Memory Allocation Algorithms: Develop shell scripts to model contiguous memory allocation methods like Best Fit, Worst Fit, and First Fit, assessing their effectiveness in memory management and fragmentation reduction.

By achieving these objectives, we aim to gain practical insights into the behavior and performance characteristics of core operating system services. The project will involve designing, implementing, and evaluating simulation scripts, followed by thorough analysis and documentation of results to draw meaningful conclusions about the simulated algorithms.

Through this project, we expect to deepen our understanding of operating system principles, enhance our scripting skills, and contribute to the broader field of computer science by providing a hands-on exploration of critical operating system functionalities.

1.5 TOOLS TECHNOLOGIES

The project is developed using the following Tools and technologies:

• **Shell Scripting:** Utilize bash scripting for implementing simulation logic.

• **Basic UNIX Commands:** Employ commands like awk, sed, grep, and sort for data manipulation.

• **Simulated Workloads:** Generate synthetic workloads to test and evaluate different algorithms.

• **Text Editor:** Description: A text editor (e.g.,online compiler, VS code, Nano) was used for writing and editing the shell script code. The text editor provided a user-friendly interface for script development and modification.

By leveraging these tools and technologies, the project team successfully implemented a shell script to simulate basic operating system services, demonstrating the versatility and practicality of shell scripting in emulating core OS functionalities.

1.6 Conclusion

This project has showcased the effectiveness of shell scripting in simulating fundamental operating system services, including process management, file handling, and resource allocation. By leveraging Bash scripting and Unix-like environments, we were able to create a functional simulation that provided insights into core OS concepts and behaviors.

Overall, simulating operating system services with shell scripting provides a valuable learning experience and practical insight into system administration and scripting technologies. This project serves as a foundation for further exploration and refinement, offering opportunities to delve deeper into OS concepts and expand the capabilities of shell scripting in system simulation and automation.the project successfully achieved

its objectives of demonstrating the capabilities of shell scripting in emulating core OS functionalities and fostering skill development in system interaction and automation.