Asia Pacific Journal of Information System and Digital Transformation
2024 , Vol 1 No 01, Research Article

Scripting Programming & Simple Messaging Between Server and Client(s) in Virtual Machine

Amirul Hakimi Bin Mohamad Nizarudin
Faculty of Computer Science in Multimedia Computing, UTHM, Parit Raja, Malaysia
amirulhakimi1110@gmail.com

Ahmad Arif Bin Ahmad Nasir
Faculty of Computer Science in Multimedia Computing, UTHM, Parit Raja, Malaysia
ariftorepride10@gmail.com

Muhamad Amirul Aiman Bin Sulehan
Faculty of Computer Science in Multimedia Computing, UTHM, Parit Raja, Malaysia
mhamirulaiman97@gmail.com

Abstract

Virtual Machine technology in this era plays a crucial role in modern Operating System environments by enabling security, flexibility and efficiency resource utilization. This study presents the implementation of scripting programming and simple messaging server and client(s) using a virtual machine based on Kali Linux. The project is divided into two main components: automated file backup and cleanup using Bash scripting, and simple client-server communication using the netcat tool. The objective of this research is to demonstrate practical operating system concepts such as automation, scheduling, file system management, and basic networking. The results show that Bash scripting can effectively automate routine administrative tasks, while netcat provides a lightweight and powerful mechanism for understanding client-server communication. This work contributes as a practical reference for students and beginners in operating system implementation using virtual machines.

Keywords : Virtual Machine, Bash Scripting, Backup Automation, Netcat, Client-Server Communication

Chapter 1 - Backup File

1.1 Introduction

In an operating system environment, file management and data protection are essential administrative tasks. Manual backup processes are prone to human error and inefficiency. Therefore, automation using shell scripting is a fundamental skill for system administrators. This chapter focuses on the development of a Bash script that automates system information logging, backup creation, and cleanup operations based on predefined conditions.

1.2 Conceptual Background

Bash scripting is a command-line scripting language commonly used in Unix-based operating systems. It allows users to automate repetitive tasks through conditional statements, loops, and system commands. In this project, conditional execution is applied to determine when backup and cleanup operations should be performed.

The backup concept implemented follows a time-based condition, where backup files are generated on specific days (Wednesday and Friday). Cleanup operations are scheduled on Saturday with user confirmation to ensure data safety.

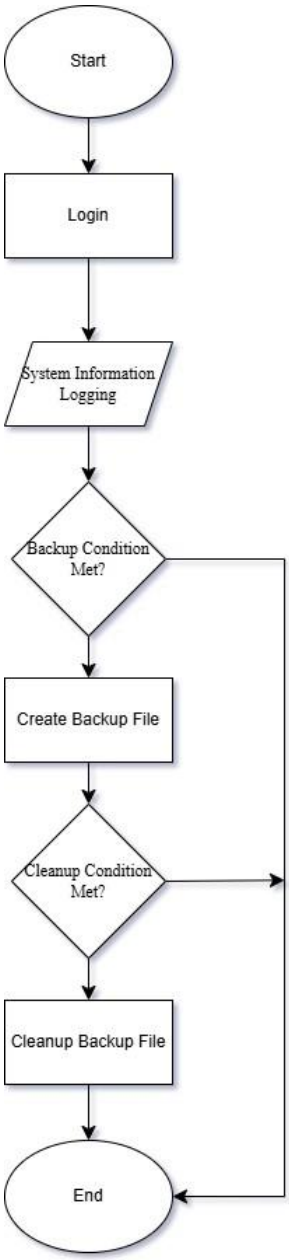

Figure 1: Conceptual flow of backup and cleanup automation

(Login → System Information Logging → Backup Condition Check → Cleanup Condition Check)

1.3 Research Methodology

The methodology for Chapter 1 follows these steps:

1. Requirement analysis based on the project specification.
2. Design of Bash scripts using conditional statements.
3. Implementation of system information logging.
4. Execution of backup and cleanup scripts.
5. Observation and documentation of outputs using screenshots.

Chapter 2 - Simple Messaging

2.1 Introduction

Networking is a core component of operating system functionality. Understanding basic client-server communication is essential for system-level networking tasks. This chapter demonstrates a simple messaging system using the netcat utility within a virtual machine environment.

2.2 Conceptual Background

The client-server model is a distributed application structure where one system (server) provides services, and another system (client) requests those services. Netcat is a networking tool that allows reading and writing data across network connections using TCP or UDP protocols.

In this project, netcat is used to simulate a simple text-based communication between two virtual machines, reinforcing the understanding of port listening, connection establishment, and data transmission.

2.3 Research Methodology

The methodology for Chapter 2 includes:

1. Configuration of two virtual machines as server and client.
2. Identification of IP address and port number.
3. Execution of netcat command on the server to listen for connections.
4. Execution of netcat command on the client to initiate communication.
5. Observation and recording of message exchange results.

Chapter 3 - Flowchart

This chapter presents the overall flow of the system implementation.

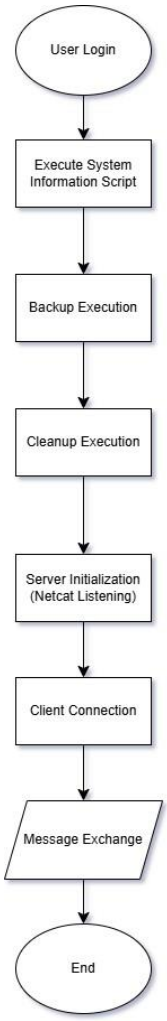

Figure 2: Overall system flowchart

1. User login to system
2. Execution of system information script
3. Conditional backup execution
4. Conditional cleanup execution
5. Server initialization

6. Client connection
7. Message exchange

The flowchart provides a clear visualization of how scripting automation and messaging tasks are integrated within the virtual machine environment.

Chapter 4 - Analysis and Development

4.1 Planning

The planning phase involved identifying the system requirements, selecting appropriate tools, and defining execution conditions. Kali Linux was chosen as the guest operating system due to its flexibility and strong command-line support. Bash scripting and netcat were selected as lightweight and effective tools for achieving the project objectives.

4.2 Development

A. Usecase Diagram

The use case diagram represents interactions between the user and the system, including script execution, backup creation, cleanup confirmation, server setup, and client messaging.

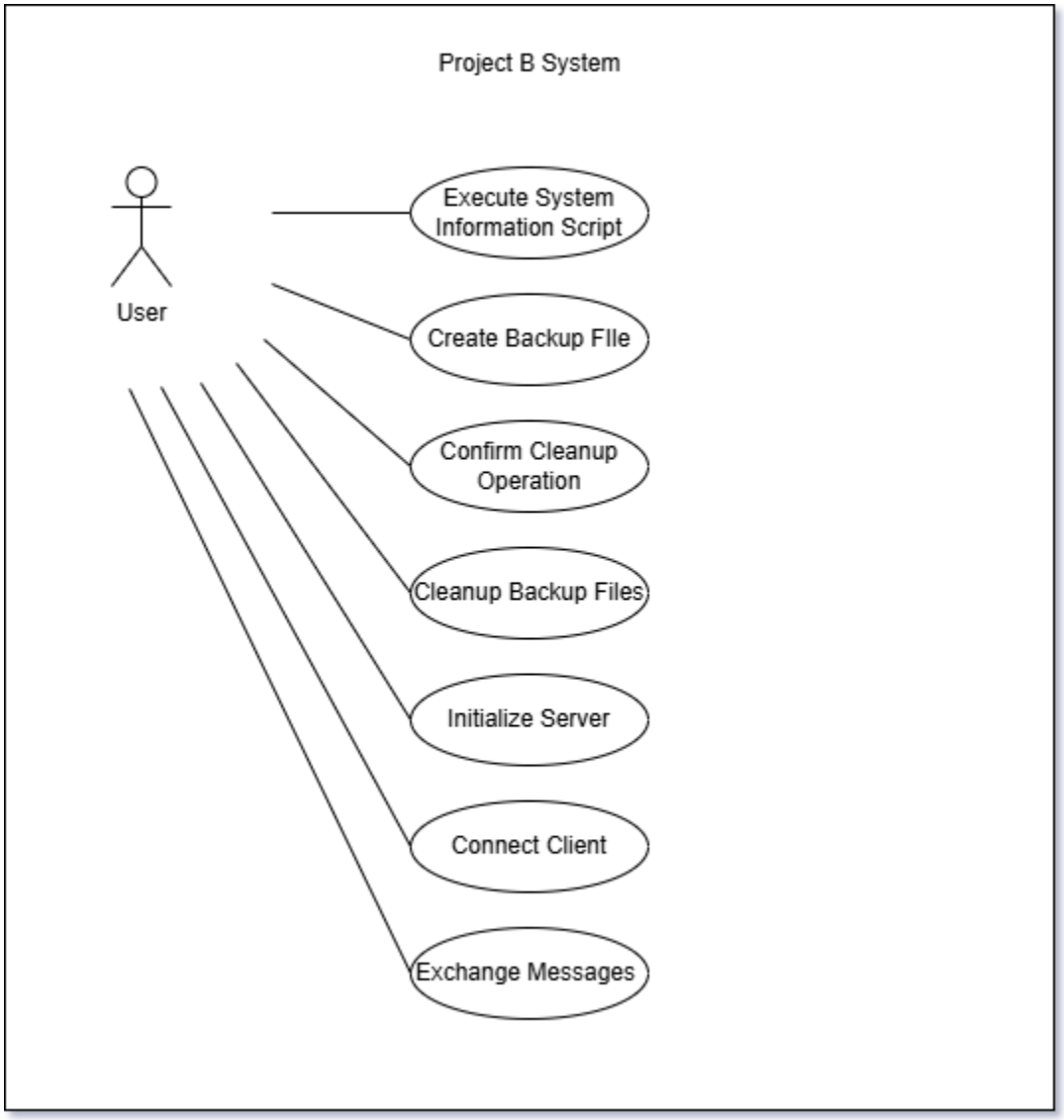

Figure 3: Use Case Diagram for Project B

B. Class Diagram

Although the project is script-based, conceptual classes such as ScriptManager, BackupHandler, and MessagingService are identified to illustrate logical separation of responsibilities.

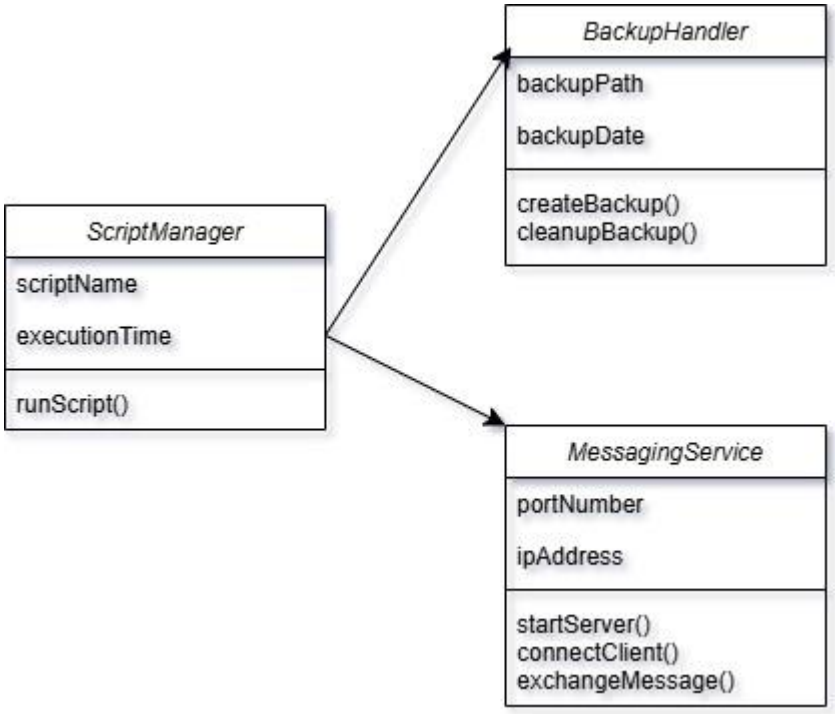

Figure 4: Conceptual Class Diagram

C. Relational Database

A lightweight conceptual database relationship is proposed to log backup metadata and communication logs for future scalability.

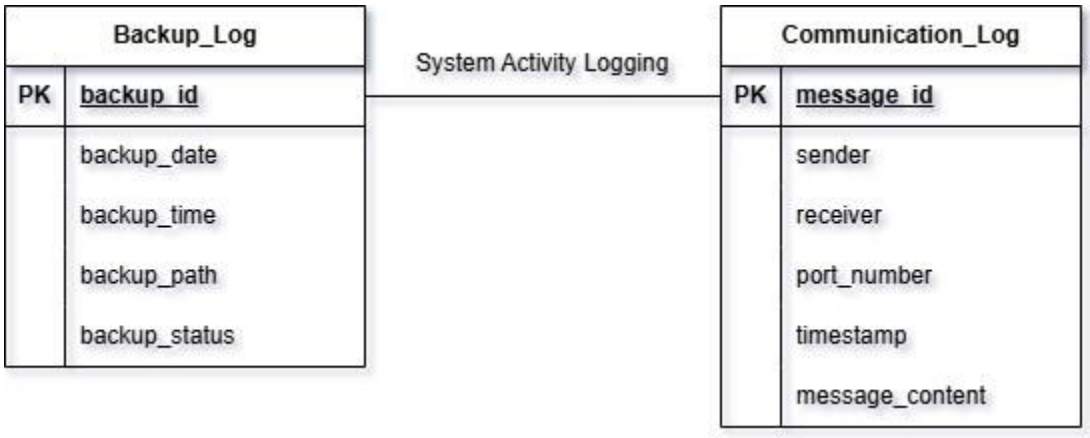

Figure 5: Conceptual Relational Database Diagram (Simple ERD)

4.3 Control

Control mechanisms are implemented through:

- User confirmation before file deletion
- Restricted port usage for messaging
- Manual termination of netcat sessions using CTRL+C

These controls ensure safe execution and prevent unintended system behavior.

Conclusion

This project successfully demonstrates the practical application of operating system concepts using a virtual machine environment. Bash scripting proved effective for automating backup and cleanup tasks, while netcat enabled a clear understanding of client-server communication. The structured approach adopted in this study aligns with academic and practical requirements, making it suitable for publication and educational reference. Future work may include scheduling using cron jobs, secure communication using encryption, and integration with graphical monitoring tools.

References

1. Shotts, W. (2019). *The Linux Command Line*. No Starch Press, from https://nostarch.com/tlcl2
2. Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems*. Pearson, from https://www.pearson.com/en-us/subject-catalog/p/modern-operating-systems/P200000003295
3. Kali Linux Documentation. (2024), from https://www.kali.org/docs/
4. Netcat Manual Page. (2024), from https://linux.die.net/man/1/nc

DOI: https://doi.org/10.61973/apjisdt.v10124.5