Lab 4

Shell Scripting and Operations​

**Introduction**

Shell scripting is a powerful way to automate tasks, execute commands, and manipulate data within a Unix-like terminal environment. Shell scripts are essentially a series of commands written in a scripting language (typically Bash) that are executed in sequence.

**Basic Scripting Constructs**

**1. Variables:**

* Declare variables: **variable\_name=value**.
* Access variables: **$variable\_name**.

**2. Comments:**

* Use **#** for single-line comments.
* For multi-line comments, enclose text within **<<COMMENT ... COMMENT**.

**3. User Input:**

* Read user input into a variable: **read variable\_name**.

**4. Conditions:**

* Use **if**, **elif**, and **else** statements for conditional execution.
* Example:

if [ condition ]; then

# code to run if condition is true

elif [ another\_condition ]; then

# code to run if another\_condition is true

else

# code to run if none of the conditions are true

fi

**5. Loops:**

* Implement loops with for and while constructs.
* Example:

for i in {1..5}; do

echo "Iteration $i"

done

**6. Functions:**

* Define functions for code reusability.
* Example:

my\_function() {

echo "Hello from my function!"

}

**Conclusion**

Shell scripting in the terminal is a versatile and essential skill for automating tasks and managing system operations. This report provides a foundational understanding of shell scripting, including script creation, execution, and basic scripting constructs. By mastering these concepts, you can create efficient and effective shell scripts to streamline your workflow and automate repetitive tasks in a Unix-like environment.