Sure! Here is a shell script that defines a function to count the number of lines in a file and calls this function with different filenames:

#!/bin/bash

# Function to count the number of lines in a file

count\_lines() {

local filename=$1

# Check if the file exists

if [ ! -f "$filename" ]; then

echo "File '$filename' not found!"

return 1

fi

# Count the number of lines in the file

local line\_count=$(wc -l < "$filename")

echo "File '$filename' has $line\_count lines."

}

# Call the function with different filenames

count\_lines "file1.txt"

count\_lines "file2.txt"

count\_lines "file3.txt"

To use this script:

1. Save it to a file, for example `count\_lines.sh`.

2. Make the script executable with the command `chmod +x count\_lines.sh`.

3. Run the script with `./count\_lines.sh`.

Make sure to replace `"file1.txt"`, `"file2.txt"`, and `"file3.txt"` with the actual filenames you want to check. This script will print the number of lines for each specified file.