CS 3307-01 Operating Systems 2

Instructor: Professor Olawale Omoyeni

Name: Anonymos

Written Assignment Unit 5

**Introduction**

In recent years, the need for automating repetitive processes in e-commerce companies has grown significantly. Efficiently managing and organizing sales data plays a crucial role in daily operations, and automating such tasks can significantly enhance productivity (Ahmed, 2020; McKay, 2022). This project aims to develop a Bash script to automatically download and organize the daily sales report (Spreadsheet-1000-rows.csv) generated by a large e-commerce company.

The key objectives of this project are as follows:

* Demonstrate how to use a Bash script to efficiently download the sales report.
* Move and organize the downloaded file into a designated folder for improved management.
* Integrate scheduling tools to automate the execution of these tasks daily.

The script is designed to be simple and extensible, ensuring that junior staff can easily set it up and execute it without difficulty. By including detailed comments and explanations for each code segment, this document provides a comprehensive understanding of how the script functions.

Following best practices for technical documentation, clear examples and step-by-step instructions have been provided to ensure transparency and reproducibility (Shotts, 2019). Through this guide, readers will be equipped to utilize the script effectively for process automation.

**Sample Script Explanation**

This section introduces a sample Bash script that downloads the Spreadsheet-1000-rows.csv file and organizes it into a designated folder. The script comprises the following functionalities:

1. File Download: Uses the `wget` command to download the CSV file from a specified URL.
2. File Organization: Moves the downloaded file to a designated directory to streamline data management.
3. Error Handling: Detects and reports errors at each step, ensuring a reliable execution.

**Sample Script**

Below is the script with detailed comments to explain its functionality:

```bash

#!/bin/bash

# Directory to store the downloaded file

TARGET\_DIR="/path/to/target/folder"

# URL of the file to download

FILE\_URL="https://example.com/valid-csv-file.csv"

# File name for the downloaded file

FILE\_NAME="data.csv"

# Create target directory if it does not exist

if [ ! -d "$TARGET\_DIR" ]; then

echo "Target directory does not exist. Creating directory..."

mkdir -p "$TARGET\_DIR"

fi

# Download the file using wget

echo "Downloading file from $FILE\_URL..."

wget -q -O "$TARGET\_DIR/$FILE\_NAME" "$FILE\_URL"

if [ $? -ne 0 ]; then

echo "Error: Failed to download the file."

exit 1

fi

# Confirm successful download

echo "File downloaded successfully to $TARGET\_DIR/$FILE\_NAME"

```

**Script Execution Steps**

Step 1: Creating the Script File

1. Use any text editor to create a new script file and save it as `download\_and\_organize.sh`.
2. Copy and paste the above code into the file.

Step 2: Granting Execution Permission

To make the script executable, run the following command:

chmod +x download\_and\_organize.sh

Step 3: Running the Script

To execute the script, use the following command:

./download\_and\_organize.sh

Step 4: Scheduling the Script (Optional)

To automate daily execution at 6 AM, add the following entry to `crontab`:

0 6 /path/to/download\_and\_organize.sh >> /path/to/logfile.log 2>&1

**Detailed Instructions**

**File Download**

The `wget` and `curl` tools can be used to download files:

* Using wget:

wget -q -O /path/to/target/folder/data.csv https://example.com/valid-csv-file.csv

* Using curl:

curl -s -o /path/to/target/folder/data.csv https://example.com/valid-csv-file.csv

**File Organization**

Ensure the directory exists, and move the file to the target location:

```bash

mkdir -p /path/to/target/folder

mv data.csv /path/to/target/folder/data.csv

```

**Error Handling**

Verify the success of each operation and handle errors accordingly:

```bash

if [ $? -ne 0 ]; then

echo "Error: Operation failed."

exit 1

fi

```

**Conclusion**

This document has outlined the process for automating the download and organization of the Spreadsheet-1000-rows.csv file using a Bash script. The script effectively achieves the following:

1. Automates file downloads using `wget` or `curl`.
2. Organizes files into designated directories for improved management.
3. Integrates with scheduling tools like `crontab` to ensure automation.

By providing error handling and detailed comments, this script is both reliable and accessible to junior staff. This solution demonstrates a practical approach to process automation, aligning with industry best practices (McKay, 2022; Ahmed, 2020).

Word Count: 626

References

1. Ahmed, H. (2020, September 2). *25 bash script examples.* FOSS Linux.
2. McKay, D. (2022, December 16). *Curl vs. wget in Linux: What’s the difference?* How-To Geek.
3. Shotts, W. (2019). *4 Manipulating files and directories. In The Linux Command Line* (5th ed., pp. 25-41). No Starch Press.