AFRICAN LEADERSHIP UNIVERSITY

CODING LAB - GROUP

BSE YEAR 1 TRIMESTER 2

LAB 1

The BSE department at ALU has partnered with an AI-based firm that utilizes satellite data to predict weather changes, providing Africa with valuable insights and solutions to optimize agricultural practices. They are collaborating with ALU to help Linux skills extract insights from the vast amounts of satellite data collected.

Your group will work with the startup to extract insights from their huge dataset on GitHub.

Task 1: Clean the repository after cloning it to your sandbox

Before we can start working on the analysis tasks, some cleanup is required.

- The directory with the name **rename_directory** should be renamed to **analyzed_data**
- Files named dummy, dummy-2, and dummy-3.txt are dummy and useless so they should be deleted/removed.
- The entire file satelite_temperature_data.csv should be inside the **raw_data** directory as this is the raw data obtained from the satellites
- The **analyzed_data** directory will contain new files that will contain the analysis conducted in Step 2

Task 2: Analyze the data with the use of Linux skills

You will use different Linux commands like cat, cp, grep, cut, sort, etc to perform analysis on the satellite dataset.

- Extract the top 10 highest temperatures from satelite_temperature_data.csv and save the result in **analyzed_data** under the name highest_temp.csv
- Extract the entire data from a country of your choice from satelite_temperature_data.csv and save the result in descending order of humidity (highest to lowest). Save the output in the **analyzed_data** directory under the name of humidity_data_$country_name.csv The $country_name represents the country name that was selected.
- **OPTIONAL Task (Bonus):** Write a shell script to extract these two insights from the file and save the outputs in the directory

SUBMISSION:

- Once the group is done, you will submit a link to your GitHub repository.
- The repository contains **analyzed_data**, **raw_data** directories, and the files containing data

AFRICAN LEADERSHIP UNIVERSITY

CODING LAB - GROUP

BSE YEAR 1 TRIMESTER 2

LAB 2

A local hospital recently upgraded its IT infrastructure for its digitization goals. This upgrade will improve patient data storage, such as heart rate data and archive logs, and securely back them up to a remote server for future access.

In this assignment, you will develop three shell scripts to manage these tasks.

Task 1: Heart Rate Monitoring Script

- **Script Name**: heart_rate_monitor.sh
    - **Description**: Write a shell script to record heart rate data every second.
    - The script should prompt for the device name (e.g., "Monitor_A", "Monitor_B", ...) and begin logging heart rate data into heart_rate_log.txt.
    - The log file should include the timestamp, Device name, and Simulated heart rate (You can use randomized data)
    - The script will run in the background, and its process ID (PID) should be displayed for management.
- **Example of content in the Log File (**heart_rate_log.txt**)**:

```
2024-10-03 14:35:02 Monitor_A 61
2024-10-03 14:35:03 Monitor_A 67
2024-10-03 14:35:04 Monitor_A 48
```

- **Expected functionality in Task 1**:
    - Run the script in the background as a separate process (i.e.: you can print the "process ID" when the script starts running).
    - View the heart_rate_log.txt to check if it is recording data every second by using the "tail -f" Linux command

Task 2: Log Archival Script

- **Script Name**: archive_log.sh
    - **Description**: Create a shell script to archive the log file heart_rate_log.txt.
    - The script archives by renaming the log file with a timestamp
    - Example of an Archived Log Name *taken on 2025 January 3rd at 14:35:01*: *heart_rate_log.txt_20241003_143501*
- **Expected functionalities in Task 2**:

    - Creating archive_log.sh script which creates an archive file of heart_rate_log.txt log file.
    - The archive is created by renaming the original heart_rate_log.txt to heart_rate_log.txt_YYYYMMDD_HHMMSS.

OPTIONAL Task (Bonus): Archival and Backup Script

- **Script Name**: backup_archives.sh
    - **Description**: After archiving the log files, you will now create a script that:
        - Moves the archived log files into a designated directory called **archived_logs_group$**. (replace the $ with your group number)
        - Back up the archived files to a remote server using SSH.
    - The script should move all archived log files (with timestamped names) into the **archived_logs_group$** directory.
    - For backup on a remote server use the SCP command via SSH.
- **Remote Backup Details**:
    - The backup script should first put all the archived log files in the archived_logs_group$ directory
    - Use a group member's sandbox as a server (sandbox) for backup and use connection details:
        - **Host:** *copy the host info of the sandbox*
        - **Username:** *copy the username of the sandbox*
        - **Directory on the backup server where to send to:** */home/*
- **Expected functionalities in Task 3**:
    - Write the backup_archives.sh script to:
        - Move all archived log files to **archived_logs_group$**.
        - Back up the directory to the remote server using SSH.
    - Ensure the script is executable and properly handles file transfers.

SUBMISSION:

You will submit a link to your GitHub repo:

- **Github repo_name:** hospital_monitoring_group$id
- The repository contains all three scripts (heart_rate_monitor.sh, archive_log.sh, and backup_archives.sh).