PROJECT REPORT

Course Code- 24CAP-607

LINUX ADMINISTRATION LAB

MASTERS IN COMPUTER APPLICATION

Submitted By:
Name- Atul Sinha
UID- 24MCA20010
Branch- MCA
Section- 1A

Submitted To :
Ms. Geetanjali Sharma

Project Title: Real-Time Clock Display Script in Bash in Linux

Introduction

This Bash script is designed to display the current time in real-time, updating every second. It uses an infinite loop to refresh the terminal display continuously. The time is shown in blue, achieved by defining ANSI color codes for terminal text formatting. This script is ideal for anyone wanting a simple, live clock display in the terminal or for beginners learning about loops, color codes, and time functions in Bash scripting.

Objective of the Script:

•**Purpose**: To display the current time in the terminal, continuously updating every second with a colored output.

Script Overview:

- **Script Language**: Bash (using #!/bin/bash)
- **Key Features**:
- Real-time time display
- Colored output (in blue)
- Continuous update every second
- Uses ANSI color codes for terminal text formatting

Components of the Script:

1. **Shell Declaration** (#!/bin/bash): Indicates that the script is written in Bash.
2. **ANSI Color Codes**: Stores color codes for displaying text in different colors (Red, Green, Blue).
3. **Infinite Loop**: Displays the current time repeatedly, refreshing every second.

Title: Detailed Code Walkthrough

1. Declaring ANSI Color Codes

```
Red=$'\e[1;31m' # ANSI code for red text
Green=$'\e[1;32m' # ANSI code for green text
Blue=$'\e[1;34m' # ANSI code for blue text
```

- **Explanation**:
- These variables define text colors for use in the terminal.
- $'\e[1;31m', $'\e[1;32m', $'\e[1;34m': ANSI escape codes to set red, green, and blue colors, respectively.
- Blue is used in this script, making the time display appear blue.

2. Continuous Loop and Display

```
while true
do
    clear # Clears the screen
    echo $Blue $(date +%T) # Displays current time in blue
    sleep 1s # Waits 1 second before updating
done
```

- **Explanation**:
- **while true**: Runs an infinite loop until the script is manually stopped (e.g., by pressing Ctrl + C).
- **clear**: Clears the terminal screen to ensure each time update appears in a clean format.
- **echo $Blue $(date +%T)**:
- echo $Blue: Sets the color of the printed text to blue.
- $(date +%T): Gets the current time in HH:MM:SS format.
- **sleep 1s**: Pauses the script for 1 second before refreshing the time display.

Title: Running the Script and Practical Applications

1. How to Run the Script

- **Save the Code**: Save the script as **clock.sh** or any filename ending with **.sh**.
- **Make it Executable**

```
chmod +xclock.s h
```

This command gives the script execute permissions.

- **Execute the Script**

```
./clock.sh
```

This starts the continuous time display in blue until manually stopped.

2.Practical Applications

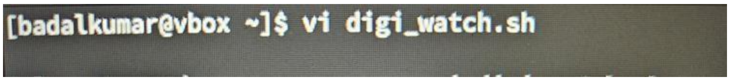
- **Real-Time Time Tracking**: Useful for monitoring time during tasks or in sessions.
- **Learning Tool**: Great example forbeginners learning Bash scripting,ANSI colors, and infinite loops.
- **System Monitoring**: Could be adapted to display other real-time data (e.g., system load, memory usage) using similar principles.

3. Stopping the Script

- To stop, use Ctrl + C, which interrupts the loop and stops the script.

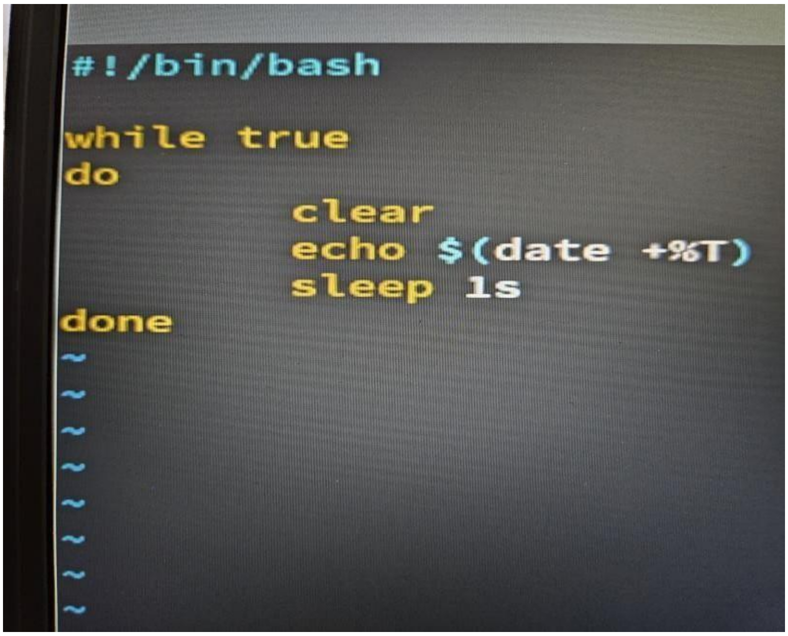
Execution:

Step 1:



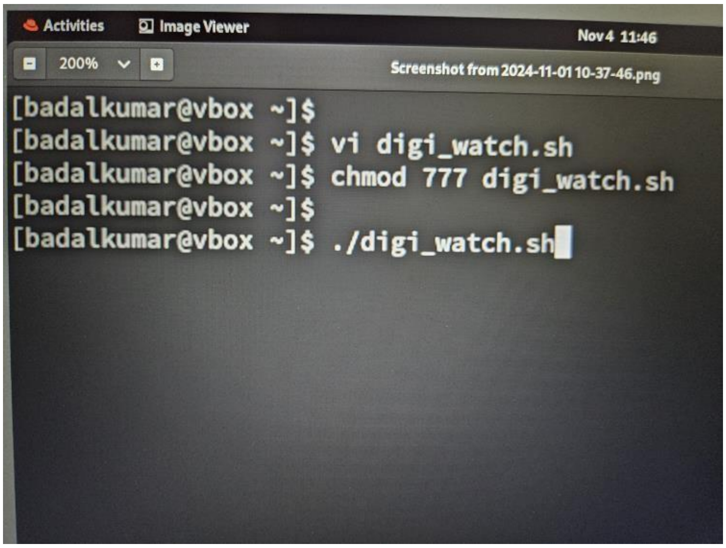
```
[badalkumar@vbox ~]$ vi digi_watch.sh
```

Step 2:



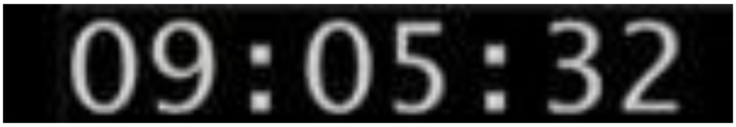
```
#!/bin/bash

while true
do
        clear
        echo $(date +%T)
        sleep 1s
done
~
~
~
~
~
~
~
~
```

Step 3:



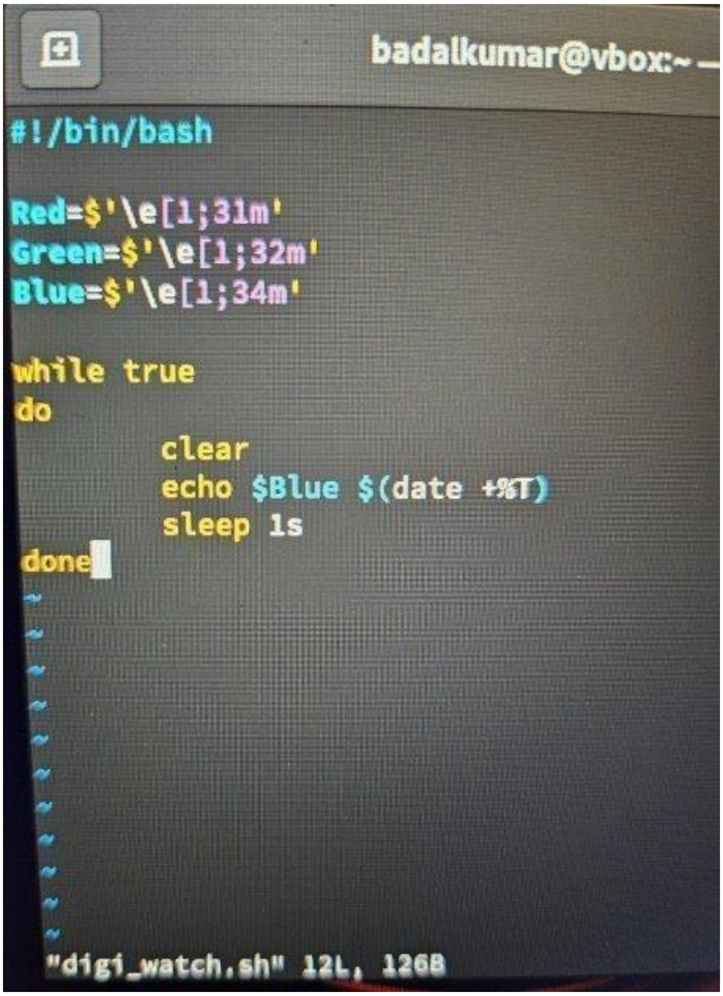
```
[badalkumar@vbox ~]$
[badalkumar@vbox ~]$ vi digi_watch.sh
[badalkumar@vbox ~]$ chmod 777 digi_watch.sh
[badalkumar@vbox ~]$
[badalkumar@vbox ~]$ ./digi_watch.sh
```

Step 4:



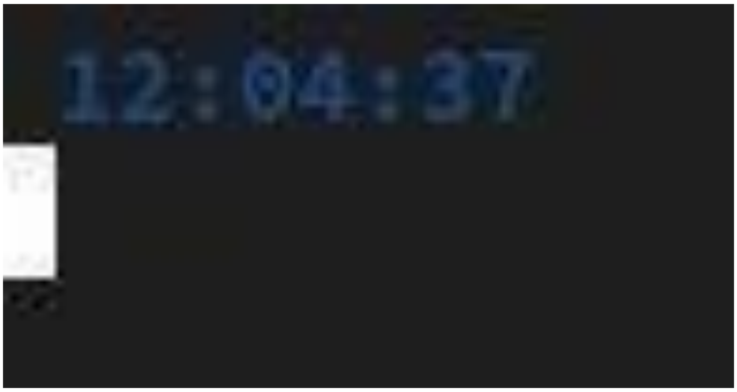
```
09:05:32
```

Step 5:



```
#!/bin/bash

Red=$'\e[1;31m'
Green=$'\e[1;32m'
Blue=$'\e[1;34m'

while true
do
        clear
        echo $Blue $(date +%T)
        sleep 1s
done
~
~
~
~
~
~
~
~
~
~
~
"digi_watch.sh" 12L, 126B
```

Step 6:



```
12:04:37
```

Conclusion:

This Bash script effectively demonstrates how to create a real-time clock display in the terminal, using loops and ANSI color codes for dynamic and visually distinct output. By utilizing an infinite loop, it continuously updates the current time every second, providing a clean and easy-to-read display. This script serves as a practical tool for monitoring time or as a learning example for understanding basic concepts in Bash, such as loops, color customization, and system commands.

Github Link:

https://github.com/Badalkumarji/Linux-mini-project

Reference:

https://www.youtube.com/watch?v=PH-rWuclGjA&t=431s