Project Report
On
Digital Clock

MASTERS OF COMPUTER APPLICATIONS

Submitted By:

Bharti Saini
(24MCA20159)

Project Guide:

Navdeep Singh Sodhi
MCA Dept., (CU)

DEPARTMENT OF COMPUTER APPLICATION,
CHANDIGARH UNIVERSITY,
(NH05, Chandigarh-Ludhiana Highway , Gharuan, Mohali , Punjab , India)

SESSION 2024-26

DECLARATION

I, Bharti Saini, hereby declare that this project report titled *" Digital Clock Using Shell Scripting"* is original work carried out by me under the supervision of Mr. Navdeep Singh Sodhi. I further declare that this work has not been submitted to any other institute/university for the award of the degree of Master of Computer Applications.

Student Name: **Bharti Saini**
Roll No: **24MCA20159**

ACKNOWLEDGEMENT

I express my sincere gratitude to my project guide, Mr. Navdeep Singh Sodhi, for invaluable guidance and support throughout this project. I also extend my thanks to Chandigarh University for the opportunity to undertake this project and to my classmates and family for their continuous encouragement.

Bharti Saini
24MCA20159

TABLE OF CONTENTS

Introduction:

This digital clock project, created using shell scripting, displays the current time in the terminal. The script uses a continuous `while` loop to update the time every second. The `clear` command keeps the display uncluttered by refreshing the screen, while `echo` combined with the `date` command shows the current time in hours, minutes, and seconds. The output color is set to blue using ANSI escape codes. This project is a simple yet effective way to demonstrate looping, color styling, and time functions in shell scripting.

Scope of the System

The digital clock system created with shell scripting has a broad yet straightforward scope. It continuously displays the current time in real-time, refreshing every second to provide an accurate digital clock within the terminal. This system is highly customizable; users can modify it to display additional information like the date, day, or various time formats, and can easily change colors or styles by adjusting the ANSI color codes. Due to its lightweight nature, this clock is resource-efficient and suitable for any Linux environment, including low-power devices. The project has strong educational value, offering a hands-on way to learn shell scripting basics, such as loops, system variables, and the `date` command. Furthermore, this simple script can be a foundation for more complex tools, allowing for the addition of features like alarms, notifications, or custom formatting for different time zones. Overall, this digital clock project provides both functional utility and an effective learning platform for script-based development.

Project Description

The Digital Clock project is a real-time clock application created with shell scripting to display the current time in a terminal. It updates every second using a continuous loop, the `date` command for fetching time, and ANSI color codes for styling. This lightweight and customizable script serves as a practical tool and an educational resource for learning shell scripting basics, and it can be expanded with features like alarms or additional formatting options.

Advantages of the Project

1 • **Real-Time Display**: Provides an accurate, continuously updating display of the current time in the terminal.

2 • **Lightweight and Efficient**: The script requires minimal resources, making it suitable for any Linux environment, including low-powered systems.

3 • **Customizable**: Users can easily modify the script to display additional information (e.g., date, day of the week) and adjust colors and formatting.

4 • **Educational Value**: Offers hands-on experience with shell scripting fundamentals, including loops, ANSI escape codes, and the `date` command.

Software and Hardware Requirements

Software:
OS: CentOS 7/8
Programming Language: Python 3.x
IDE: Visual Studio Code / PyCharm
Libraries: PyQt5 (for future GUI implementation)

Hardware:
Processor: Intel i5 or higher
RAM: 4 GB minimum
Disk Space: 50 GB minimum

Project Development Stages

- **Planning and Requirements Gathering**: Define the project objectives, such as creating a real-time clock display. Determine the required features, like continuous updating, color customization, and simplicity of design.

- **Environment Setup**: Ensure that the development environment is configured with a Linux terminal and a text editor. Familiarize yourself with basic shell commands and test the environment to confirm compatibility with ANSI escape codes and the `date` command.

- **Script Design**: Outline the structure of the script, including the use of a `while` loop for continuous operation, the `clear` command to refresh the display, and ANSI color codes for output styling. Plan any variables, such as colors or time formats.

- **Implementation**: Write the script by coding each component:

  - Set up color variables using ANSI codes.
  - Use a `while true` loop to make the script run indefinitely.
  - Clear the screen and display the current time using the `date` command.
  - Add a `sleep` interval to update the time every second.

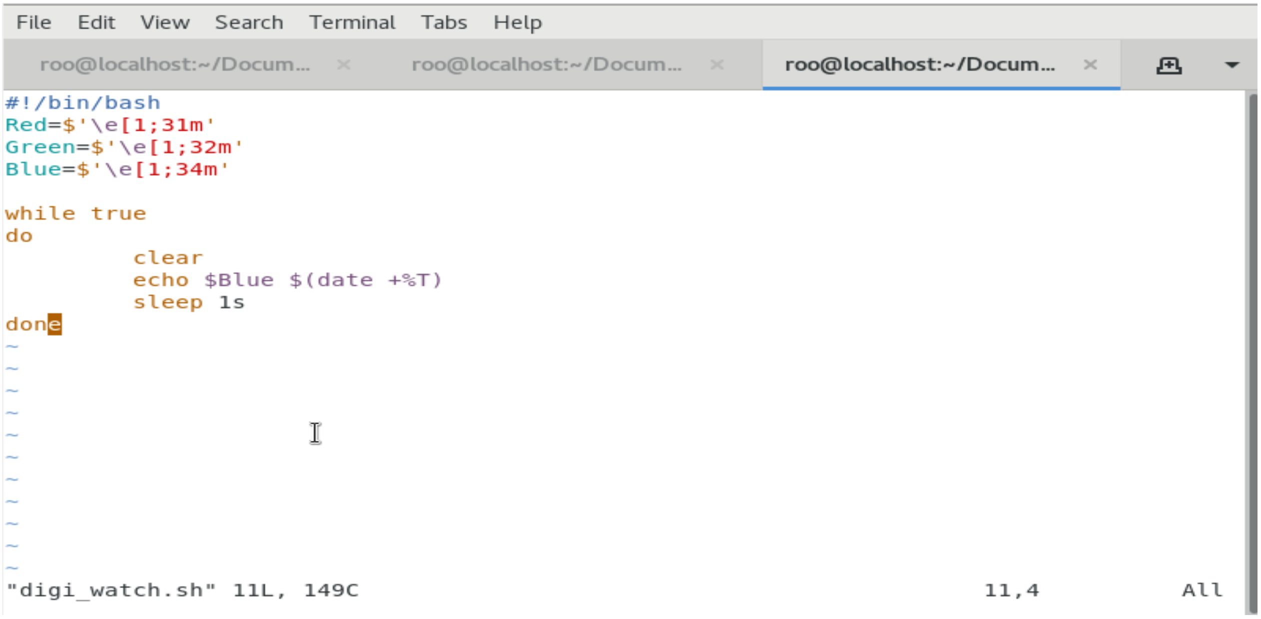
- **Testing and Debugging**: Run the script in the terminal to ensure it displays the time correctly and refreshes every second. Adjust any issues with formatting, timing, or color display, and verify that the script works smoothly.

- **Customization and Enhancement**: Add optional features, such as showing the date or customizing the color scheme. Experiment with different time formats or additional functionalities, like alarms or notifications.

- **Documentation**: Create a brief guide explaining the purpose, usage, and customization options of the script. Include comments within the script for easier understanding and maintenance.

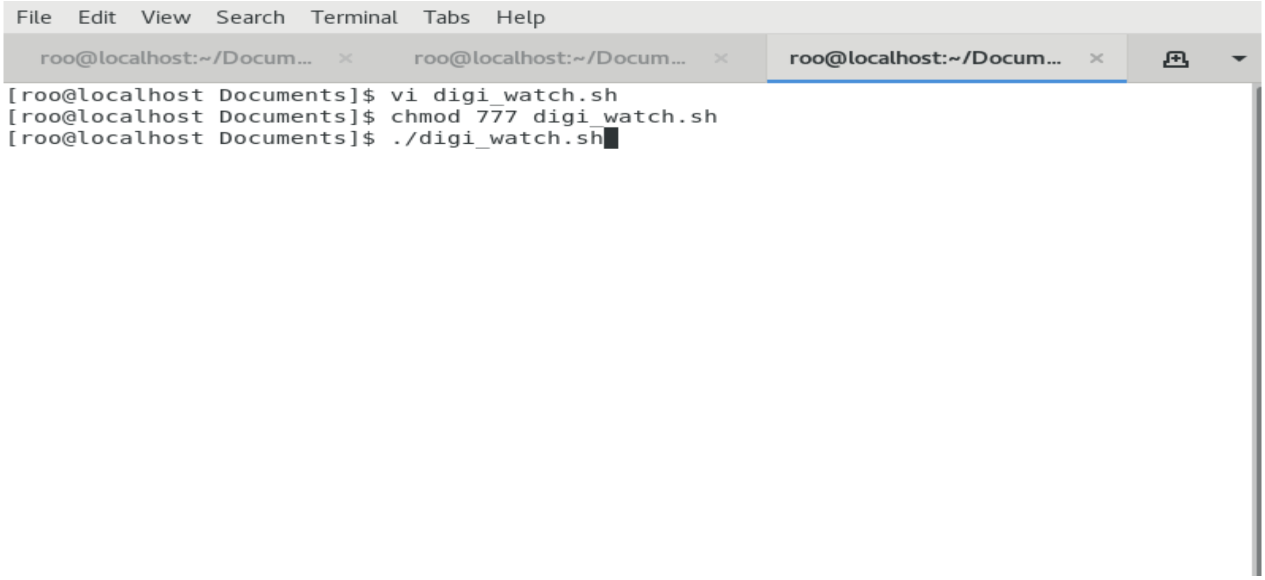
- **Deployment and Maintenance**: Deploy the script for use in any desired environment. Regularly check for any compatibility issues with updates to the shell environment, and modify the script as needed for new features or optimizations.

Screenshots



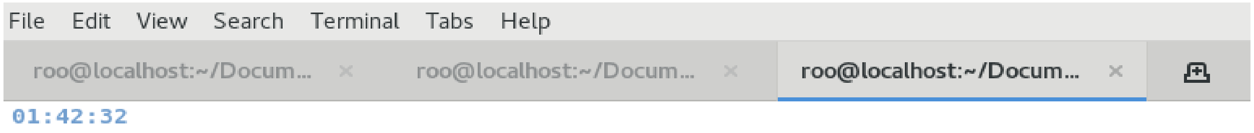
```
#!/bin/bash
Red=$'\e[1;31m'
Green=$'\e[1;32m'
Blue=$'\e[1;34m'

while true
do
          clear
          echo $Blue $(date +%T)
          sleep 1s
done
~
~
~
~
~
~
~
~
~
~
~
"digi_watch.sh" 11L, 149C                                         11,4          All
```



```
[roo@localhost Documents]$ vi digi_watch.sh
[roo@localhost Documents]$ chmod 777 digi_watch.sh
[roo@localhost Documents]$ ./digi_watch.sh
```

Output:



```
01:42:32
```

Future Enhancements

1.Date and Day Display: The full date and day of the week with customizable time zones for more diverse usability is allowed.

2.Custom Time Formats: Enables 12-hour or 24-hour format or adds milliseconds for a more detailed clock.

3.Color and Style Themes: Offer the ability to have different themes and styles for the display of the clock, such as options for fonts, text effects (blinking or bold text), and background colors.

4.World Clock: The multiple time zones or world clock view, so users can see what time it is in various cities around the world.

5.Interactive User Interface: Permit interactive setting up of any preferences in the menu, keyboard shortcut, keyboard controls that would include any form of toggle on a display date, time formatting, and a alarm among others.

6.Logging and Analytics: There could be choices of time or usage patterns in logs periodically which might aid other purposes for time recording.

7.Battery and System Monitoring: Hook up system tools so it can display some information, like the battery level, the CPU usage, or even network status, and it morphs into a mini-dashboard

8.Graphical Interface: create a GUI version, using some tools like Zenity or GTK, letting one visually interact with this clock for users who lack direct experience in the command line.

Conclusion

The Digital Clock is the project that makes shell scripts practical, educational, and configurable for showing real-time information about time in the terminal of Linux. Loops, color coding, real-time updating are all basics on shell scripting that would give a great deal in displaying this kind of work with beginners. Moreover, due to its light nature, it is more than likely to work anywhere, including low-power system environments. There are several scopes to make the project for future enhancements, varying from adding alarms and world clocks to creating a graphical user interface. Overall, it is an excellent foundational exercise in scripting combined with a base for further development in applications through the terminal.

Bibliography

1. **Bash Guide for Beginners**: A beginner-friendly guide to Bash scripting basics. Bash Guide
2. **The Linux Command Line**: A comprehensive book covering Linux command line essentials. Linux Command Line
3. **GNU Coreutils Manual**: Detailed documentation on core utilities like `date` and `clear`. Coreutils
4. **Advanced Bash-Scripting Guide**: An in-depth resource for advanced Bash scripting techniques. Advanced Bash-Scripting