



UNIVERSITY INSTITUTE *of*
COMPUTING
Asia's Fastest Growing University

NAAC
GRADE A+
ACCREDITED UNIVERSITY



CHANDIGARH
UNIVERSITY

Discover. Learn. Empower.

Linux: Project Report

**Submitted for partial fulfillment of requirement for the award
of**

Degree of

Master of Computer Applications

Chandigarh University, Mohali

Submitted By:

Name: Rahul Kumar

U.ID: 24MCA20169

Submitted To:

Ms. Prabhjot Bhatti Mam

Digital Clock Using Shell Scripting

1.Introduction

This project involves creating a digital clock using shell scripting on Linux CentOS. Shell scripting is a powerful tool in Unix-like operating systems, allowing automation of various tasks. For this project, the script is designed to display the current time in a real-time, continuous format on the terminal, updating every second.

2.Objectives

The main objective of this project is to:

- Develop a shell script that displays a real-time digital clock.
- Update the time display every second to mimic the behavior of a standard digital clock.
- Use basic shell scripting commands and tools available on CentOS, such as `date`, `sleep`, and `loops`.

3.Prerequisites

To run this project, ensure that the following components are available on your system:

- **CentOS Linux** installed (any recent version).
- Basic understanding of **Bash shell scripting**.
- Familiarity with terminal and command-line interface.

4. Implementation

Tools Used

- **Shell Scripting (Bash):** The script is written in Bash, the default shell in CentOS.
- **date Command:** To fetch the current time.
- **sleep Command:** To create a delay of 1 second between each update.
- **clear Command:** To refresh the display for the updated time.

5.Script Explanation

The following steps are taken in the implementation:

1. **Get the Current Time:** The `date` command is used to fetch the current time in the desired format.
2. **Clear the Screen:** The `clear` command is used to refresh the terminal screen for each time update.

3. **Looping:** A `while` loop is used to ensure that the clock continuously runs and updates every second.
4. **Delay:** The `sleep` command introduces a 1-second pause between each update to simulate the ticking of a clock.

6.Explanation of the Script

- `#!/bin/bash`: This line tells the system to use Bash as the interpreter for the script.
- `while true`: Creates an infinite loop that continues until the user stops the script manually.
- `clear`: Clears the terminal screen before each time update to create a clean display.
- `date +"%H:%M:%S"`: Fetches the current time in Hour:Minute
- `format`.
- `sleep 1`: Pauses the execution of the script for 1 second to ensure the clock updates in real time.

7. How to Run the Script

Step 1: Prepare the Environment

Before starting, ensure you have access to a CentOS system. Bash is the default shell on CentOS, so no additional software is needed beyond the terminal.

Step 2: Create the Script File

We will create a shell script file to write the clock logic. Follow these steps to create a script file.

1. Open the terminal.

```
rahul@localhost ~]$ date
```

2. Use a text editor like `vi` or `nano` to create a new shell script file. For example, use `vi` to create the file named `digital_clock.sh`:

```
rahul@localhost ~]$ vi digi_watch.sh
```

The editor will open, and you can begin writing your script.

3. Now, write the script that will display the digital clock in real-time.

`#!/bin/bash` at the start of the script tells the system that this script should be interpreted using the Bash shell.

```
#!/bin/bash
```

4. Create an Infinite Loop: Since we want the clock to continuously update, we use a while true loop that runs indefinitely.

```
while true  
do
```

5. Clear the Terminal Screen: To make the clock display clean, we clear the screen before updating the time. This ensures that the new time replaces the old one, preventing overlap.

```
clear
```

6. Fetch and Display the Current Time: The date command is used to fetch the current system time. To display the time in HH:MM

format, we use the +"**%H:%M:%S**" option of the date command.

```
$(date +%T)
```

7. Pause for 1 Second: After displaying the time, we need the script to wait for 1 second before updating again. This is done using the sleep command.

```
sleep 1s
```

8. Close the Loop: The done command ends the while loop, ensuring it repeats indefinitely.

```
done
```

9. Complete Script

Here is the final version of the script, combining all the steps above

```
#!/bin/bash  
  
while true  
do  
    clear  
    echo $(date +%T)  
    sleep 1s  
done
```

Save and Exit

To save the script in vi editor, press Esc, then type :wq and press Enter. This saves the file and closes the editor.

Run the Script

Now that the script is executable, you can run it by typing the following command in the terminal:

```
[rahul@localhost ~]$ ./digi_watch.sh
```

Stop the Script

Since the script is an infinite loop (it will run indefinitely), you need to stop it manually when you no longer need it. To stop the script, press **Ctrl + C** in the terminal. This will break the loop and terminate the script.

Output

```
File Edit View Search Terminal Help
18:29:51
```

Color code to change terminal text apply

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

echo $Green $(date +%T)
```

Final Output

```
File Edit View Search Terminal Help
18:43:10
```

Explanation of Key Components

Here's a breakdown of the important commands and components used in the script:

- **#!/bin/bash**: The shebang line tells the system to use Bash to interpret the script.
- **while true**: This is an infinite loop that runs the commands inside it continuously.
- **clear**: Clears the terminal screen, ensuring a clean display for each clock update.
- **date +%H:%M:%S**: Fetches and displays the current time in hours, minutes, and seconds format.
- **sleep 1**: Pauses the script for 1 second between each loop iteration, making it act like a real clock.
- **Ctrl + C**: The keyboard shortcut to terminate the script manually.

Conclusion

By following these steps, you have successfully created a digital clock using shell scripting on **Linux CentOS**. The script utilizes basic shell commands (date, clear, sleep, and loops) to continuously display the current time in real-time. This project demonstrates the power of shell scripting and how it can be used to automate tasks and create simple, useful utilities.

This project demonstrates how shell scripting can be used to create a functional digital clock that continuously displays the current time on Linux CentOS.