

CST1500

Computer Systems Architecture and Operating Systems

Bash Project

By

Shayan Ahmed(M00983522), Fawaaz Shaikh(M00909151)

Script1

#!/bin/bash

# Function to display date and time

datetime() {

datetimee=$(date +"%Y-%m-%d %H:%M:%S")

dialog --infobox "Current Date and Time:\n$datetimee" 8 40

sleep 2

}

# Function to display calendar and add information

calendar() {

# Display a calendar and get the selected date

dialog --calendar "Select a date to add information:" 0 0 2>/tmp/calendar\_result

selecteddate=$(cat /tmp/calendar\_result)

# If a date is selected, prompt for additional information

if [ -n "$selecteddate" ]; then

dialog --inputbox "Add relevant information for $selecteddate:" 8 40 2>/tmp/info\_result

info=$(cat /tmp/info\_result)

echo "Information for $selecteddate: $info"

sleep 2

fi

}

# Function to delete a file

deletefile() {

# Prompt for the directory path or use the current directory

dialog --inputbox "Enter the directory path (leave blank for current directory):" 8 40 2>/tmp/dir\_result

directory=$(cat /tmp/dir\_result)

[ -z "$directory" ] && directory="."

# Get the list of files in the specified directory

files=$(ls "$directory")

# Display files with a numbered list

file\_list=""

index=1

for file in $files; do

file\_list+=" $index \"$file\""

((index++))

done

# Prompt the user to select a file for deletion

dialog --menu "Select a file to delete:" 0 0 0 $file\_list 2>/tmp/delete\_result

selected\_index=$(cat /tmp/delete\_result)

# If a file is selected, confirm deletion

if [ -n "$selected\_index" ]; then

selected\_file=$(ls "$directory" | sed -n "${selected\_index}p")

dialog --yesno "Are you sure you want to delete $selected\_file?" 8 40

response=$?

# If the user confirms, attempt to delete the file

if [ $response -eq 0 ]; then

full\_path="$directory/$selected\_file"

rm -i "$full\_path" # Use -i for confirmation

delete\_status=$?

# Check if deletion was successful

if [ $delete\_status -eq 0 ]; then

dialog --infobox "$selected\_file deleted successfully." 8 40

sleep 2

else

dialog --msgbox "Error deleting $selected\_file." 8 40

fi

fi

fi

}

# Main menu

while true; do

# Display the main menu and get the user's choice

menu\_choice=$(dialog --menu "Select an option:" 0 0 0 \

1 "Date/Time" \

2 "Calendar" \

3 "Delete" \

4 "Exit" \

2>&1 >/dev/tty)

# Act based on the user's choice

case $menu\_choice in

1) datetime ;;

2) calendar ;;

3) deletefile ;;

4) exit ;;

\*) dialog --msgbox "Invalid choice. Please try again." 8 40 ;;

esac

done

script2

#!/bin/bash

# Function to display system information

function showsystem {

echo "1) Operating System Type:"

echo "$(lsb\_release -sd)"

echo -e "\n2) Computer CPU Information:"

echo "$(lscpu)"

echo -e "\n3) Memory Information:"

echo "$(free -h)"

echo -e "\n4) Hard Disk Information:"

echo "$(lsblk)"

echo -e "\n5) File System (Mounted):"

echo "$(df -h)"

}

# Display information

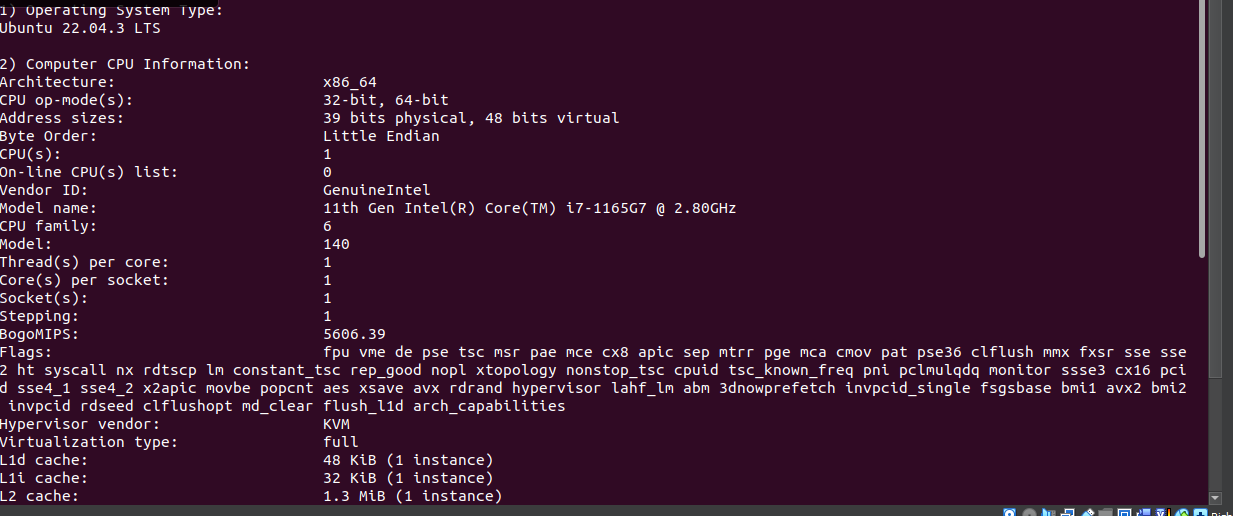
showsystem

Output of script1

A screenshot of a computer program

Description automatically generated

Output of script2



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Self-report

Report by Shayan: In working with both Bash scripts, I've acquire a comprehensive understanding of Bash scripting principles. The first script focuses on interactive functionalities, employing dialog-based interfaces and modular functions to manage date/time, calendar events, and file deletion effectively. Through this, I've learned about code modularity, user interaction, error handling, and dynamic data manipulation within a terminal environment.

The second script emphasizes system information retrieval and display, utilizing functions and command-line tools to gather details on the operating system, CPU, memory, disks, and file systems. This script underscores the importance of function-based organization, command substitution for formatting, and automation for system insights dissemination.

By combining these experiences, I've deepened my knowledge of Bash scripting fundamentals, including modularity, user interaction, error handling, and system administration automation. These skills are valuable for creating efficient, user-friendly scripts capable of automating various system management tasks seamlessly.

Report by Fawaaz : With the help of this project I got to learn about linux, ubuntu and bash in brief. Earlier I was familiar with neither Linux nor with bash. But now I hold a concrete understanding of what Linux is and what Bash is. I learned about how to create new virtual machines. I also learned that you can control your whole system with the help of commands. It was amazing to see how some set of commands can give you the power of controlling the whole system. I  learned to create a dialog box and in that we tried to display different options. I also learned about the calendar function. Overall I learned a lot about shell scripting. Shell scripting in Windows and in Linux both was fun. I now have a brief understanding of how to set up a virtual machine with the help of Oracle VM VirtualBox. After download VirtualBox I also needed a “DISC” file with the extension of “.ISO” which was nothing but ubuntu’s installation file. It was not easy at first as what to do with the DISC file but later when I referred to some tutorials and articles I got to know that you need to upload that DISC file in VirtualBox whilst setting up your Virtual Machine for Ubuntu.