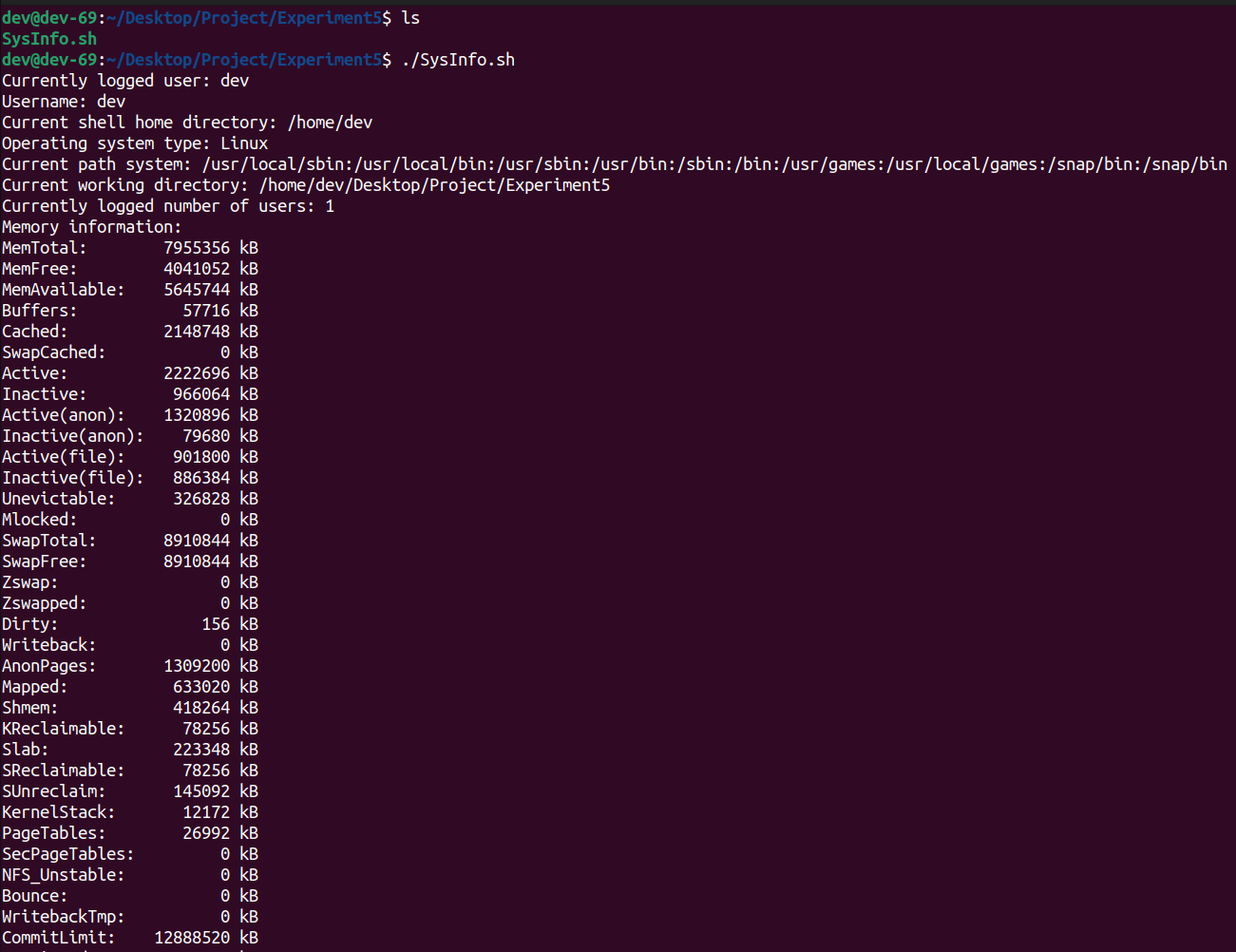
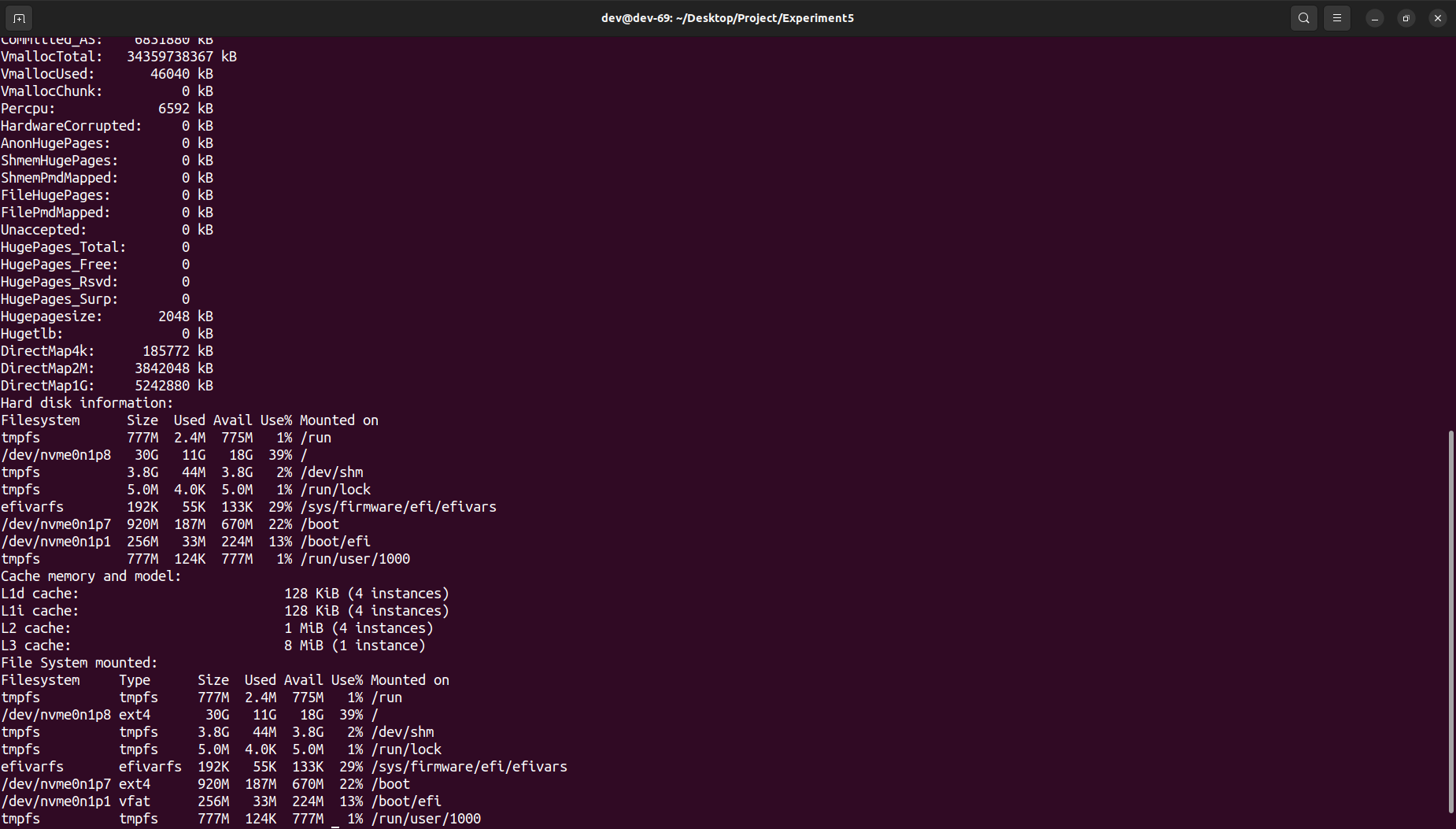
Q1.Write a shell script to show various system configuration like currently logged user and his logname, your current shell, home directory, operating system type, current path setting, current working directory, show currently logged number of users, show memory information, Hard disk information like size of hard-disk, cache memory, model etc, and file system mounted





Q2.Write a shell script to add user and password on Linux system

#!/bin/bash

if [ "$EUID" -ne 0 ]; then

echo "Please run this script with sudo or as root."

exit 1

fi

read -p "Enter username: " username

if id "$username" &>/dev/null; then

echo "User '$username' already exists. Please choose a different username."

exit 1

fi

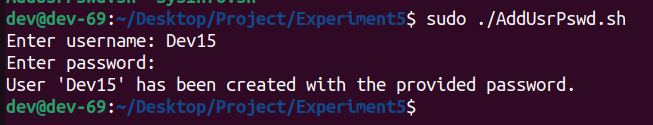
read -s -p "Enter password: " password

echo

useradd -m "$username"

echo "$username:$password" | chpasswd

echo "User '$username' has been created with the provided password."



Q3.Write a shell script to print last login details and Number of users logged in

#!/bin/bash

# Print last login details for all users

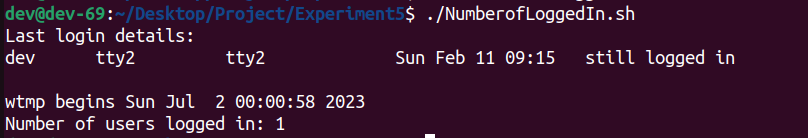
echo "Last login details:"

last -1

# Print number of users currently logged in

echo -n "Number of users logged in: "

who | wc -l



Q4.Write a shell script to upgrade and cleans the system automatically instead of doing it manually

#/bin/bash

echo "Updating package lists"

sudo apt update

echo "Upgrading installed packages"

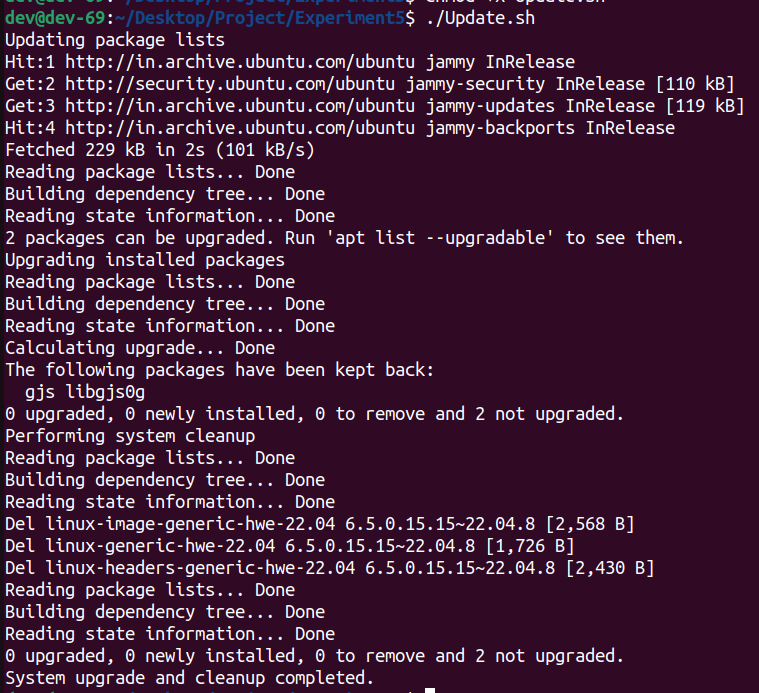
sudo apt upgrade -y

echo "Performing system cleanup"

sudo apt autoclean

sudo apt autoremove -y

echo "System upgrade and cleanup completed."



Q5.Write a shell script to delete all log files present inside your var/log directory

#!/bin/bash

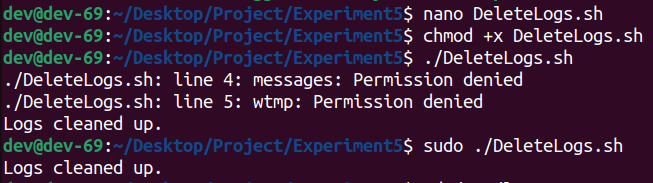
LOG\_DIR=/var/log

cd $LOG\_DIR

cat /dev/null > messages

cat /dev/null > wtmp

echo "Logs cleaned up."



Q6. Write script to check if user is root user

#!/bin/bash

ROOT\_UID=0

if [ "$UID" -eq "$ROOT\_UID" ]

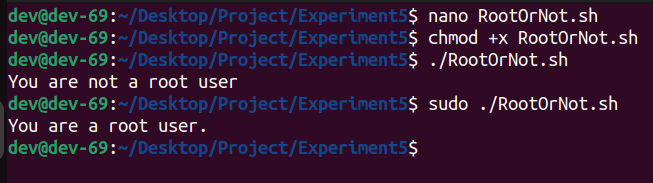
then

echo "You are a root user."

else

echo "You are not a root user"

fi



Q7. Write script to print number of files and folders

#by default find includes the starting directory

#!/bin/bash

if [ -d "$@" ]; then

echo "Files found: $(find "$@" -type f | wc -l)"

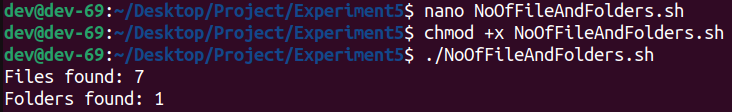
echo "Folders found: $(find "$@" -type d | wc -l)"

else

echo "[ERROR] Please try again."

exit 1

fi



Q8.Write a script that accepts the hostname and IP address as command-line arguments and adds them to the /etc/hosts file.

#!/bin/bash

if [ "$EUID" -ne 0 ]; then

echo "Please run this script as root"

exit 1

fi

if [ $# -ne 2 ]; then

echo "Usage: $0 <hostname> <IP address>"

exit 1

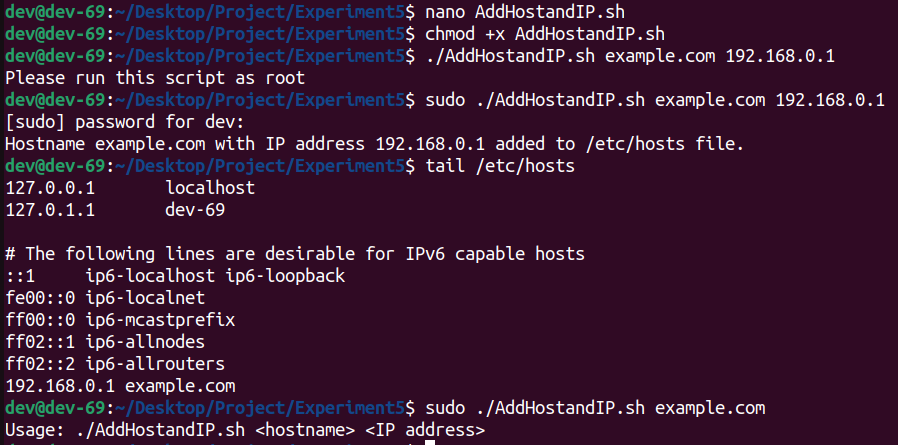
fi

hostname="$1"

ip\_address="$2"

echo "$ip\_address $hostname" >> /etc/hosts

echo "Hostname $hostname with IP address $ip\_address added to /etc/hosts file."



Q9. Write Script to Create Directories

#!/bin/bash

echo -n "Enter directory name ->"

read dir

if [ -d "$dir" ]

then

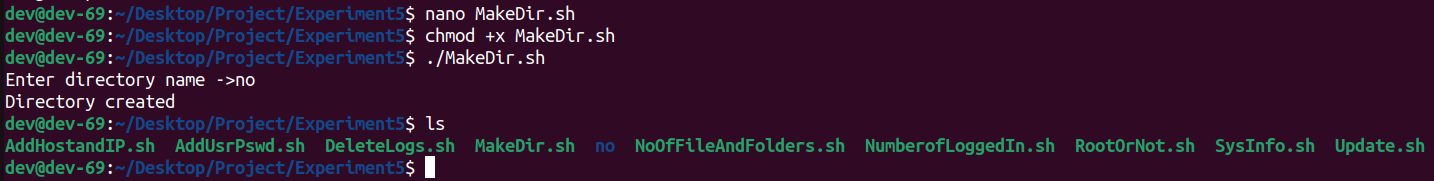
echo "Directory exists"

else

`mkdir $dir`

echo "Directory created"

fi



Q10. Write shell script to demonstrate Command Line Argument

i.Positional Parameters

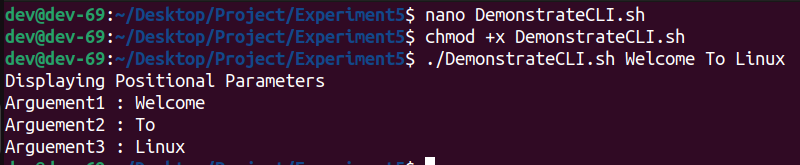
#!/bin/bash

echo "Displaying Positional Parameters"

echo "Argument1 : $1"

echo "Argument2 : $2"

echo "Argument3 : $3"



ii.

#!/bin/bash

echo "Displaying Flags"

while getopts "1:2:3:" flag; do

case "${flag}" in

1) argument1=${OPTARG} ;;

2) argument2=${OPTARG} ;;

3) argument3=${OPTARG} ;;

\*) echo "Unknown flag: ${flag}" ;;

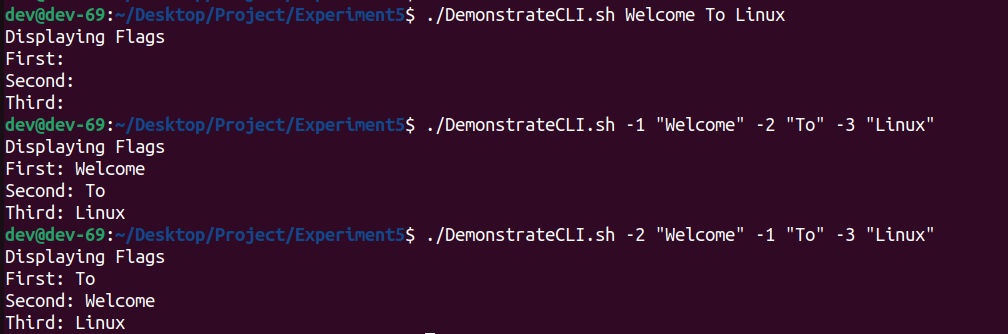
esac

done

echo "First: $argument1"

echo "Second: $argument2"

echo "Third: $argument3"



iii. Using Loops with $@ – Loop Constructs

#!/bin/bash

echo "Using Loop Constructs"

i=1

for argument in "$@"

do

echo "Argument $i : $argument"

i=$((i+1))

done

