**Linux custom Scipt Tasks**

**solution 1)**

**step 1** ) #!/bin/bash

function display\_help() {

echo "Usage: internsctl [OPTIONS]"

echo "Custom Linux Command for Operations"

echo ""

echo "Options:"

echo " --help"

echo " --version"

}

function display\_version() {

echo "internsctl v0.1.0"

}

case "$1" in

--help) display\_help ;;

--version) display\_version ;;

\*) echo "Invalid option. Use 'internsctl --help' for usage guidelines." exit 1 ;;

esac

then Save the script in a file named **'internsctl.sh'**

**step2)** to Make the script executable by running:

chmod +x internsctl.sh

**step 3)** Run the script with either --help or --version options:

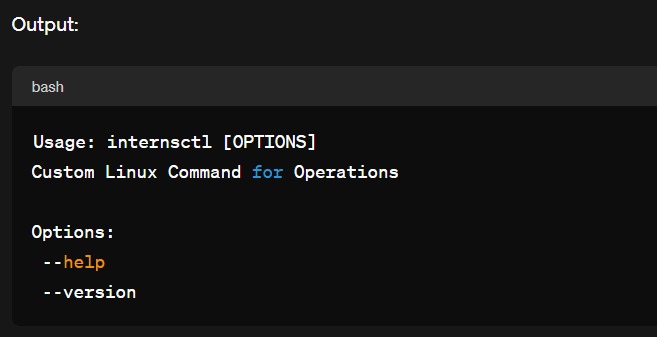
./internsctl.sh --help

or

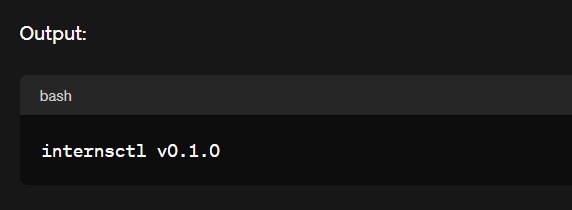
./internsctl.sh –version

**output:**

if we apply **--help** then we get



and we if apply **–version** then we get



**solution 2) part 1)**

**step 1)**

#!/bin/bash

function display\_help() {

echo -e "\e[43mUsage: internsctl [OPTIONS]\e[0m"

echo -e "\e[43mCustom Linux Command for Operations\e[0m"

echo ""

echo -e "\e[43mOptions:\e[0m"

echo -e "\e[43m --help\e[0m"

echo -e "\e[43m --version\e[0m"

}

function display\_version() {

echo -e "\e[43minternsctl v0.1.0\e[0m"

}

function get\_cpu\_info() {

lscpu

}

function get\_memory\_info() {

free

}

case "$1" in

--help) display\_help ;;

--version) display\_version ;;

cpu)

if [ "$2" == "getinfo" ]; then

get\_cpu\_info

else

echo "Invalid subcommand for 'cpu'. Use 'internsctl cpu getinfo'."

exit 1

fi

;;

memory)

if [ "$2" == "getinfo" ]; then

get\_memory\_info

else

echo "Invalid subcommand for 'memory'. Use 'internsctl memory getinfo'."

exit 1

fi

;;

\*)

echo "Invalid option. Use 'internsctl --help' for usage guidelines."

exit 1

;;

esac

then Save the modified script in a file named **internsctl.sh.**

**step 2)** to Make the script executable:

chmod +x internsctl.sh

**step 3)** Run the script with the cpu getinfo and memory getinfo commands:

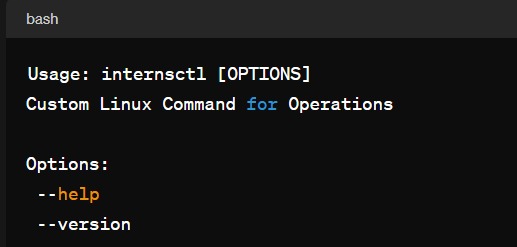
./internsctl.sh cpu getinfo

and

./internsctl.sh memory getinfo

**output:**

Running ./internsctl.sh --help:



and

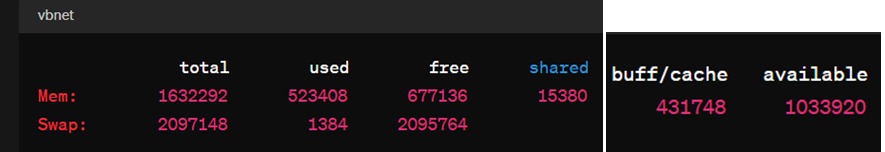
Running ./internsctl.sh –version:

then Running ./internsctl.sh cpu getinfo



and Running ./internsctl.sh memory getinfo:

it shows



**Part 2)level intermediate**

**solution:**

**step 1)**

#!/bin/bash

function display\_help() {

echo " user create Create a new user"

echo " user list"

echo " user list --sudo-only"

}

function create\_user() {

if [ -z "$2" ]; then

echo "Error: Missing username. Usage: internsctl user create <username>"

exit 1

fi

sudo useradd -m "$2"

echo "User '$2' created successfully."

}

function list\_users() {

cut -d: -f1 /etc/passwd

}

function list\_sudo\_users() {

getent group sudo | cut -d: -f4 | tr ',' '\n'

}

case "$1" in

user)

if [ "$2" == "create" ]; then

create\_user "$@"

elif [ "$2" == "list" ]; then

if [ "$3" == "--sudo-only" ]; then

list\_sudo\_users

else

list\_users

fi

else

echo "Invalid subcommand for 'user'. Use 'internsctl user create <username>' or 'internsctl user list [--sudo-only]'."

exit 1

fi

;;

\*)

echo "Invalid option. Use 'internsctl --help' for usage guidelines."

exit 1

;;

esac

then Make the script executable:

**step 2)** chmod +x internsctl.sh

**step 3)** Run the commands:

./internsctl.sh user create testuser

This will create a new user named testuser. Enter the sudo password when prompted.

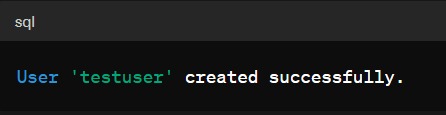
./internsctl.sh user list

This will list all users on the system.

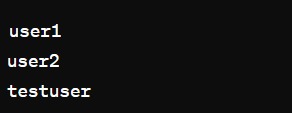
./internsctl.sh user list --sudo-only

**Output:**

Running ./internsctl.sh user create testuser:



Running ./internsctl.sh user list:



Running ./internsctl.sh user list --sudo-only:



**Part3 | Advanced Level**

**Solution:**

**step 1)**

#!/bin/bash

function display\_help() {

echo "Usage: internsctl file getinfo [OPTIONS] <filename>"

echo "Get information about a file."

echo ""

echo "Options:"

echo " --size, -s Print file size"

echo " --permissions, -p Print file permissions"

echo " --owner, -o Print file owner"

echo " --last-modified, -m Print last modified time"

}

function get\_file\_info() {

local file="$1"

local size=$(stat -c %s "$file")

local permissions=$(stat -c %A "$file")

local owner=$(stat -c %U "$file")

local last\_modified=$(stat -c %y "$file")

echo "File: $file"

echo "Access: $permissions"

echo "Size(B): $size"

echo "Owner: $owner"

echo "Modify: $last\_modified"

}

function get\_specific\_info() {

local file="$1"

local option="$2"

case "$option" in

--size|-s) echo $(stat -c %s "$file") ;;

--permissions|-p) echo $(stat -c %A "$file") ;;

--owner|-o) echo $(stat -c %U "$file") ;;

--last-modified|-m) echo $(stat -c %y "$file") ;;

\*) echo "Invalid option. Use 'internsctl file getinfo --help' for usage guidelines." ; exit 1 ;;

esac

}

if [ "$1" == "--help" ]; then

display\_help

exit 0

fi

if [ "$#" -lt 2 ]; then

echo "Error: Missing filename. Usage: internsctl file getinfo [OPTIONS] <filename>"

exit 1

fi

filename="${!#}" # Get the last argument as the filename

if [ ! -f "$filename" ]; then

echo "Error: File '$filename' not found."

exit 1

fi

if [ "$#" -eq 2 ]; then

get\_file\_info "$filename"

else

option="${2}"

get\_specific\_info "$filename" "$option"

fi

**step 2)** This script uses the stat command to retrieve file information. Save this script to a file, make it executable (chmod +x filename.sh), and then you can run it as described in your example:

./internsctl.sh file getinfo hello.txt

./internsctl.sh file getinfo --size hello.txt

./internsctl.sh file getinfo --permissions hello.txt

./internsctl.sh file getinfo --owner hello.txt

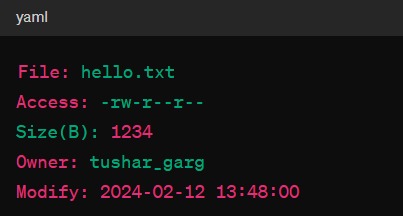
./internsctl.sh file getinfo --last-modified hello.txt

**step 3)** The script also provides a help message when --help is used:

./internsctl.sh file getinfo –help

**output:**

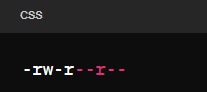
Running ./internsctl.sh file getinfo hello.txt



Running ./internsctl.sh file getinfo --size hello.txt:



Running ./internsctl.sh file getinfo --permissions hello.txt:



Running ./internsctl.sh file getinfo --owner hello.txt:

C:\Users\abc\Downloads\WhatsApp Image 2024-02-12 at 1.50.57 PM.jpeg

Running ./internsctl.sh file getinfo --last-modified hello.txt:

