Linux Fundamentals

INTRO TO CYCBER

STUDENT: MUHAMMAD FEROZ (S10)

Scope

- Automation to display Linux Operating system information using Bash scripting.
- Display Linux version
- Display Private/Public IP address and Default gateway
- Display Hard disk size, free and used space.
- Display top 5 directories and their size.
- Display CPU usage with 10-sec refresh rate.

Content Page

Bash Script Layout

Header

Figliet command is used as the introduction to the project, followed by the author of the script.

```
#Banner introduction
figlet -mini Project - LINUX Fundamentals
echo " ******* S10 - Muhammad Feroz **********
```

Variables

Variables for the select loop have to be defined, each variable carries a string of text related to the scope of the project. The variables are passed to the select statement to be listed as a numeric choice for the user to interact with the script.

1. version
2. IP
3. disk
4. directory
5. cpu

```
# Variables to store string to be displayed in Select loop
version="Display Linux version"
IP="Display private, public and default gatway"
disk="Display the hard disk size, free and used space"
directory="Display the top 5 directories and their size"
cpu="Display the CPU usuage, refresh every 10 sec"
```

Select Loop

Variable **num** of the select loop is used to refer to each variable sequenced by space. Each time the loop executes the variable **num**, it passes through the list defined earlier. After every selection, a list of commands will be executed according to the appropriate selection cases.

```
select num in "$version" "$IP" "$disk" "$directory" "$cpu" "Exit_Menu" #listing all variables
do
  case $num in "$version")
```

After the execution of a set of commands, the selection loop continues until the **Exit_Menu** choice is selected to **break** the loop, or the loop is ended manually with the CTRL+C keys.

```
Exit_Menu)
      break # To exit the selection loop
```

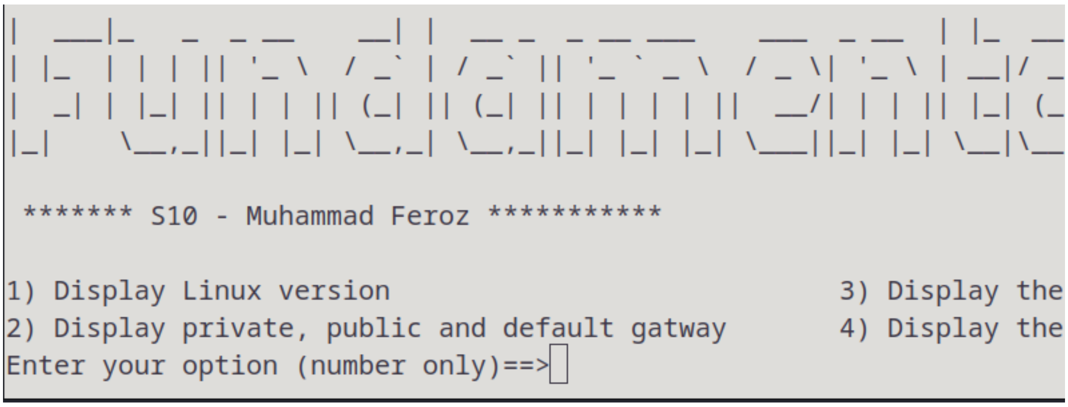
Each case displays formation after the **echo** command with the **-n** flag. This prints other commands output the same line of the echo's argument.

```
echo -n "Version: ";lsb_release -a |grep -i "Release:" | awk '{print $2}'        #grep only the value and customise the output 1 colu
```

Prompt statement (PS3)

PS3 statement is used to define a custom prompt for a select loop within the script. The statement will be displayed below the numbered selections of the loop. This option will be more intuitive for the end user instead of displaying the symbol "#$" by default provided by the select statement.

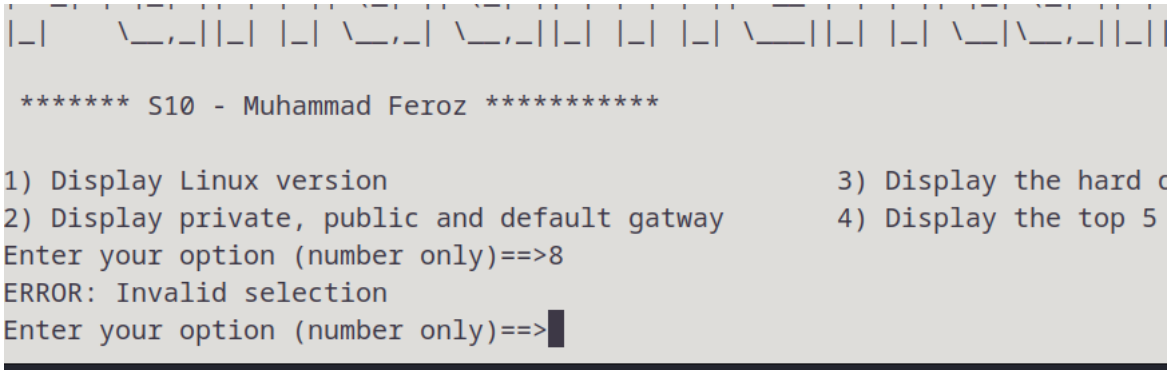
```
PS3="Enter your option (number only)==>"
```



Validation of loop

The last case out of the 5 cases contains validation and the **num** variable checks against the keyed-in value by the user. This is denoted by a "*". Incorrect selection will trigger a message and prompt the user to re-enter the value.

```
*) echo "ERROR: Invalid selection" # Any other option apart from the 6 choices is repeat the loop
```



Conditions of Cases

A total of 5 cases feed into the selection loop and each case will execute its respective commands according to the scope.

Case: Version

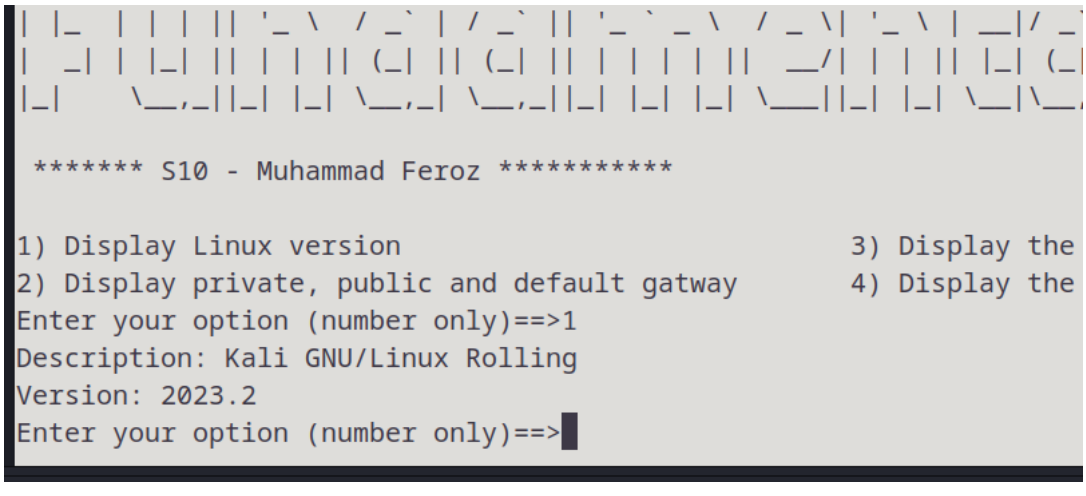
This case displays the Linux description and version of the host. The **lsb_release** command was selected as it uses the information from both **/etc/os-release** and **/usr/lib/os-release**.

lsb_release

Flag **-a** is used to show all information within the **lsb_release** command. **Grep** command is used to gather the information lines related to the "Description" and the "Release" only.

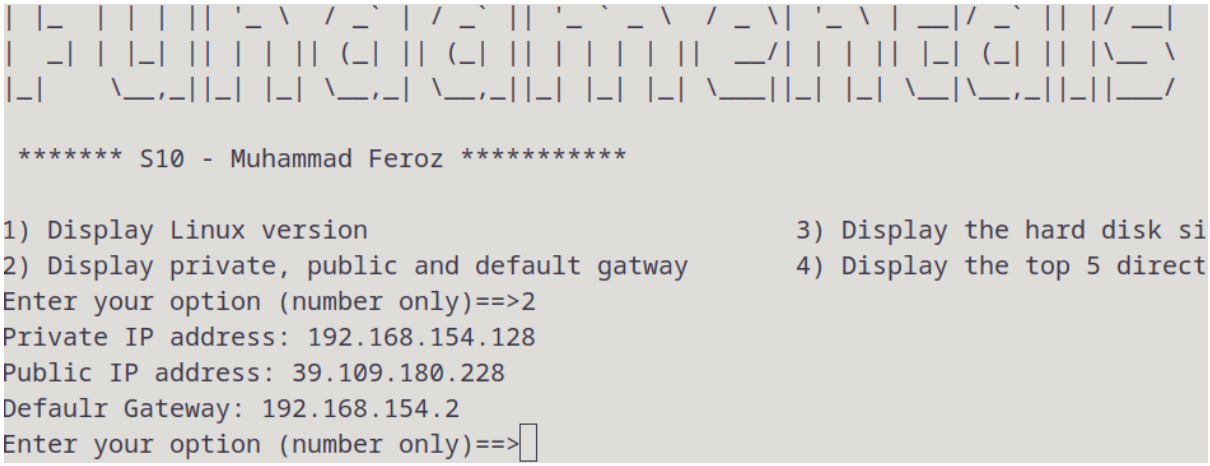
Final values were extracted using the **awk** commands and displayed.

```
case $num in "$version")
  echo -n "Description: "; lsb_release -a |grep "Description:" | awk '{print $2, $3 ,$4}' #grep only the value and customise the outpu
  echo -n "Version: ";lsb_release -a |grep -i "Release:" | awk '{print $2}'         #grep only the value and customise the output 1 co
  ip
  ;;
```



Case: IP

This case displays the private, public and default gateway of the host IP address. The **hostname**, **curl** and **ip** commands were selected to achieve the outcome.



hostname

This command displays the system's DNS name and with the -I flag it list all the network address of the host.

```
echo -n "Private IP address: ";hostname -I
```

curl

This command is used to transfer data to or from a server. It complimented with ifconfig.me, a web service that displays information of the host connection.

```
echo -n "Private IP address: ";hostname -I
```

ip

This command is used to show routing/network devices and with the passing **route** as an object, it returns the routing table entry. **Grep** command is used to gather the information lines related to the "Default" only.

Final values were extracted using the **awk** commands and displayed.

```
echo -n "Defaulr Gateway: ";ip route | grep default | awk '{print $3}'
```

Case: disk

This case displays the host's hard disk size, free and used space. The **df** command was selected to achieve the outcome of displaying the amount of space on the file system. **-h** flag is used to print sizes in power of 1024 to be easily understood.

```
 ******* S10 - Muhammad Feroz ***********

1) Display Linux version                              3) Display the hard disk size, fre
2) Display private, public and default gatway         4) Display the top 5 directories a
Enter your option (number only)==>3
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1        79G   14G   61G  19% /
Total Hard disk space: 79G
Availabe Hard disk space: 61G
Used Hard disk space: 14G
Enter your option (number only)==>
```

A **/dev-sda** augment is passed to this command. "**/**" refers to the root and "**dev**" is a is a directory containing all the device files within the root. "sda" will be the first disk of the host.

> 💡 The current host is managed by a single disk, df -H /dev/sd* can be used to manage a host with multiple disks.

Grep command is used to gather the information lines related to the "Size", "Avail" and "Used" only.

Final values were extracted using the **awk** commands and displayed.

```
df -h /dev/sda1
     echo -n "Total Hard disk space: ";df -h /dev/sda1 | awk '{print $2}'|grep -v 'Size' #grep value from 2nd column and customise the
     echo -n "Availabe Hard disk space: ";df -h /dev/sda1 | awk '{print $4}'|grep -v 'Avail' #grep value from 4th column and customise
     echo -n "Used Hard disk space: ";df -h /dev/sda1 | awk '{print $3}'|grep -v 'Used'  #grep value from 3rd column and customise the
```

Case: directory

This case displays the host's top 5 directories and their sizes. The **du** command was selected to achieve the outcome of displaying the amount of estimated file space usage. **-h** flag is used to print sizes in a human-readable format to be easily understood.

```
 ******* S10 - Muhammad Feroz ***********

1) Display Linux version                          3) Display the hard disk size, free and used space  5) Dis
2) Display private, public and default gatway     4) Display the top 5 directories and their size     6) Exi
Enter your option (number only)==>4
Top 5 directories and their size:
14G     /
12G     /usr
6.0G    /usr/lib
4.5G    /usr/share
1.9G    /usr/lib/x86_64-linux-gnu
Enter your option (number only)==>
```

A **/ 2>/dev/null** augment is passed to this command. "**/**" refers to executing the command from the root directory and "2" is to pick out all standard errors of the **du** command and inject using "**>**" command into a special virtual device called "**/dev/null**".

This “**/dev/null**” helps to discard any values sent to it, thus preventing the displaying of errors such as permission denied to certain directories.

The **sort** command is used to sort lines of text, where its complimented with the flag **-h** for more readable number sort and flag **-r** for to descending order.

The top 5 results are picked using the **head** command with **-n5** has an option.

```
echo "Top 5 directories and their size: ";du -h / 2>/dev/null |sort -h -r |head -n5 # search from root, error msg passed to temp folde
```

Case: cpu

This case displays the host’s CPU usage with a refresh rate of every 10 seconds. The **sar** command was selected to achieve the outcome as it is also known as the CPU utilization report command. The **-u** flag is used to print a report of CPU utilization with all the CPU fields and **-h** flag is used for human-readable format.

The interval of value 10 is set for this command, to get the latest CPU utilization after every 10 seconds and the time stamp on the first column confirms this.

```
|  _| | |_| || | | || (_| || (_| || | | | | ||  __/| | | || |_| (_| || |\__ \
|_|    \__,_||_| |_| \__,_| \__,_||_| |_| |_| \___||_| |_| \__|\__,_||_|_||___/

 ******* S10 - Muhammad Feroz ***********

1) Display Linux version                          3) Display the hard disk size, free and used space  5) Display the CPU usuage, refresh every 10 sec
2) Display private, public and default gatway      4) Display the top 5 directories and their size     6) Exit_Menu
Enter your option (number only)==>5
Please provide the number of iteration for this report : 5
CPU usuage, refresh every 10 sec: Linux 6.1.0-kali9-amd64 (kali) 08/31/2023

02:42:55 AM CPU %user
02:43:05 AM all 2.1%
02:43:15 AM all 1.2%
02:43:25 AM all 6.6%
02:43:35 AM all 1.3%
02:43:45 AM all 1.2%
Average: all 2.5% 0.0%
Enter your option (number only)==>
```

💡 Challenge: The user is required to perform a manual exit using the CTRL+C command to stop the ‘live report’. this in turn has a negative impact on the select loop causing a total exit from the script.
Workaround: The user is required to provide a maximum iteration (variable **countA**) for this command to prevent the ‘infinite’ loop and exit to the main select loop (menu)

```
    echo -n "Please provide the number of iteration for this report : "           # to prevent manual exit which affect the select loo
    read countA
    echo -n "CPU usuage, refresh every 10 sec: "; sar -h -u 10 "$countA" | awk '{print $1,$2,$3,$4;}' #10 sec refresh of values
```

Script

#! /bin/bash

#Banner introduction
figlet -mini Project - LINUX Fundamentals
echo " ******* S10 - Muhammad Feroz ***********
"

Variables to store string to be displayed in Select loop

```
version="Display Linux version"
IP="Display private, public and default gatway"
disk="Display the hard disk size, free and used space"
directory="Display the top 5 directories and their size"
cpu="Display the CPU usuage, refresh every 10 sec"

# PS3 custom prompt for select loop
PS3="Enter your option (number only)==>"
select num in "$version" "$IP" "$disk" "$directory" "$cpu" "Exit_Menu" #listing all variables
do
	case $num in "$version")
	echo -n "Description: "; lsb_release -a |grep "Description:" | awk '{print $2, $3 ,$4}' #grep only the value and customise the output from 3 columns
	echo -n "Version: ";lsb_release -a |grep -i "Release:" | awk '{print $2}'		#grep only the value and customise the output 1 column
	;;
		"$IP")
		echo -n "Private IP address: ";hostname -I 		#use flag to get the value and customise the output for hostname
		echo -n "Public IP address: ";curl ifconfig.me 		#curl to get the value and customise the output for IP add
		echo ""
		echo -n "Defaulr Gateway: ";ip route | grep default | awk '{print $3}'		#grap default and customise the output for Gateway from 3rd column

	;;
		"$disk")
		 df -h /dev/sda1
		 echo -n "Total Hard disk space: ";df -h /dev/sda1 | awk '{print $2}'|grep -v 'Size' #grep value from 2nd column and customise the output for total space
		 echo -n "Availabe Hard disk space: ";df -h /dev/sda1 | awk '{print $4}'|grep -v 'Avail' #grep value from 4th column and customise the output for avail space
		 echo -n "Used Hard disk space: ";df -h /dev/sda1 | awk '{print $3}'|grep -v 'Used'	#grep value from 3rd column and customise the output for used space

	;;
		"$directory")
		echo "Top 5 directories and their size: ";du -h / 2>/dev/null |sort -h -r |head -n5 # search from root, error msg passed to temp folder and result sorted and top 5 displayed
	;;
		"$cpu")
		echo -n "Please provide the number of iteration for this report : "		# to prevent manual exit which affect the select loop
		read countA
		echo -n "CPU usuage, refresh every 10 sec: ";	sar -h -u 10 "$countA" | awk '{print $1,$2,$3,$4;}' #10 sec refresh of values

	;;
		Exit_Menu)
			break # To exit the selection loop
	;;
	*) echo "ERROR: Invalid selection" # Any other option apart from the 6 choices is repeat the loop

	;;

	esac
done
```