

Task 3

By : **Ziad Ahmed Ibrahim**

Required: A script that is executed to print which WIFI network has the highest strength

Listing Methods I found

1) Using the iw tool for listing

- a tool to manipulate wireless devices
- iwlist was the method to be used to list devices
- `iwlist [interface] scan` command is used to display some information from wireless network interface
- shows ESSID, Quality, Frequency, Mode.
- can only be accessed by root

```
Cell 01 - Address: 88:03:55:E8:3A:D0
          Quality=23/70  Signal level=-87 dBm
          Encryption key:on
          ESSID:"VGV7519E83ADB"
          IE: WPA Version 1
              Authentication Suites (1) : PSK
          IE: IEEE 802.11i/WPA2 Version 1
              Authentication Suites (1) : PSK
Cell 02 - Address: 88:03:55:E8:3A:D1
          Quality=23/70  Signal level=-87 dBm
          Encryption key:off
          ESSID:"KPN Fon"
Cell 03 - Address: 90:5C:44:C5:B8:9D
          Quality=59/70  Signal level=-51 dBm
          Encryption key:on
          ESSID:"Doggie"
          IE: WPA Version 1
```

```
Authentication Suites (1) : PSK
IE: IEEE 802.11i/WPA2 Version 1
Authentication Suites (1) : PSK
Cell 04 - Address: 54:FA:3E:60:F9:B1
Quality=22/70 Signal level=-88 dBm
Encryption key:on
ESSID:"HZN249093067"
IE: IEEE 802.11i/WPA2 Version 1
Authentication Suites (1) : PSK
```

Output Example of iwlist

Command I used

```
output=$(iwlist wlan0 scan | awk -F':=' '/ESSID/{ssid=$2}
/Quality/{print ssid ":" $2}')
```

- Description
 - List wifi networks available
 - use delimiters : or = to separate fields
 - when you recognize the word "ESSID" store the second field in ssid
 - When you find quality append "ssid" and ":" and '\$2' denoting second field after quality

Other method

- You can also parse the data by looping through networks but in my opinion the one above is better

2) Using nmlci

- it is a tool that should be manually installed to your distribution
- using the command `nmlci device wifi` you can list all the wifi networks
- you can store the output from the above command in a variable
- and then use the command `nmcli device wifi connect` to connect directly to the best wifi network

Script to print the best network

- Method used is looping through the data
- using the ':' as delimiter
- using if condition to get the greatest value
- store the name and max quality in 2 variables
- echo the best network if present
- else echo error message

```

output="W2:150
W3:75
W4:65
W5:55
:49
:30
WE2F1E55:27
we30:25"
#Now we need to parse this list into name and strength of signal

max_signal=0                #Variable to hold the maximum signal
                              initialized with 0
best_network=""              #variable to hold the name of the network

#First we declare the internal field separator as the ":" in which
#contents before the IFS are of another type from contents after it
#so we loop through the list by this while loop
while IFS=: read -r network signal; do
#we loop and store the data before the : in network
#store the data after the : in signal
#We used -r as read command to treat the "/" as literal chars not a
part of a path

    if [[ -n $network ]]; then                #Check if
the signal name is empty or not for example like ""
        echo "Network: $network - Signal: $signal"    #In this
line we print the network names and strength

#in the following line we check if the signal of the network being
examined is greater than the signal in the variable best_signal
        if [[ $signal -gt $max_signal ]]; then
            max_signal=$signal                #if true store the
signal here

```

```

        best_network=$network                                #and store the
network name here
    fi
fi
done <<< "$output"      # In this line we tell the loop to use the
list provided to us "OUTPUT"

if [[ -n $best_network ]]; then                            #If we obtained a working
network
    echo "Recommended Network: $best_network"
else    #if no networks detected
    echo "Can't recommend any WIFI network"
fi

```

Output obtained

The screenshot shows a terminal window with the following content:

```

#!/bin/bash
#By Ziad Ahmed

#####
# Method 1 #
#####
#To list the wifi networks we use the iwlist command and
#for example iwlist (wlan0 or wlan1) scan
#the above command lists everything about the networks b
#we first store the networks in a variable
#like this output1=$(iwlist wlan0 scan)
#Then we can parse it by looping through the contents of
#storing the contents of the $ESSID:Quality inside anothe
#and then increment the variable each time we input a ne
#incrementing acts like a /n
#for example we can store the name like this
#if [[ $line =~ "ESSID:" ]]; then
#    name=$(echo "$line" | cut -d '"' #since we know
#####
# Method 2 #####Better#####
#####
#output=$(iwlist wlan0 scan | awk -F[:=]' '/ESSID/{ssid=
#this method is intended to list all networks with their
#using more complex structures inside the command in whic
#we extract names and strength using awk
#more information about awk can be found in my pdf
#we use piplining to acheive this
#So we the command above goes through the steps
#1)Get the list of the wifi networks nearby
#2)Use the characters ':' or '=' as a delimiter to differentiate between fields
#3)When you find the part "ESSID" take the field after delimiter using "$2" and store it in ssid
#4)Then when you find the part "Quality" append ":" and the second field after it
#5)Doing this gives us Output= Network_name:Network_quality for all networks

```

The terminal output shows the following results:

```

Network: W2 - Signal: 150
Network: W2 - Signal: 75
Network: W2 - Signal: 65
Network: W2 - Signal: 55
Network: WE2F1E55 - Signal: 27
Network: we30 - Signal: 25
Best network to connect to: W2
msi@msi-VirtualBox:~$ ./network.sh
Network: W2 - Signal: 150
Network: W3 - Signal: 75
Network: W4 - Signal: 65
Network: W5 - Signal: 55
Network: WE2F1E55 - Signal: 27
Network: we30 - Signal: 25
Best network to connect to: W2
msi@msi-VirtualBox:~$

```

Sources

hyperlinked

- [iw tool](#)

- [While loop in bash](#)
- [Bash flags](#)
- [awk parsing tool](#)
- [nmlci](#)
- [Parsing data in linux](#)