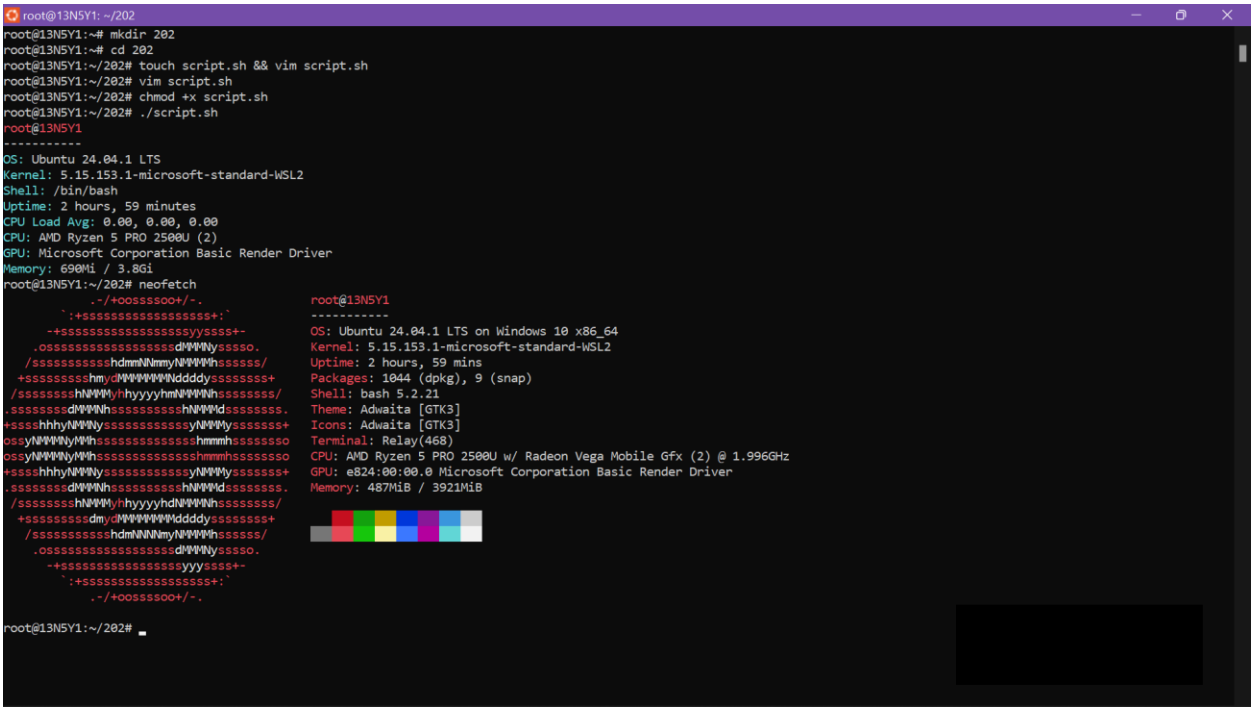
Objective: Create a Linux health monitoring script that reports key system metrics, such as disk space, memory usage, CPU load, running processes, and network connectivity. This will provide a real-time snapshot of the system's health. Follow the instructions below to complete the assignment and submit a detailed report, including screenshots of your implementation

Implementation: Created a replica of 'Neofetch' a popular package used to obtain information of *NIX (Linux, MacOS, FreeBSD etc.) systems.

Screenshots:

Output:



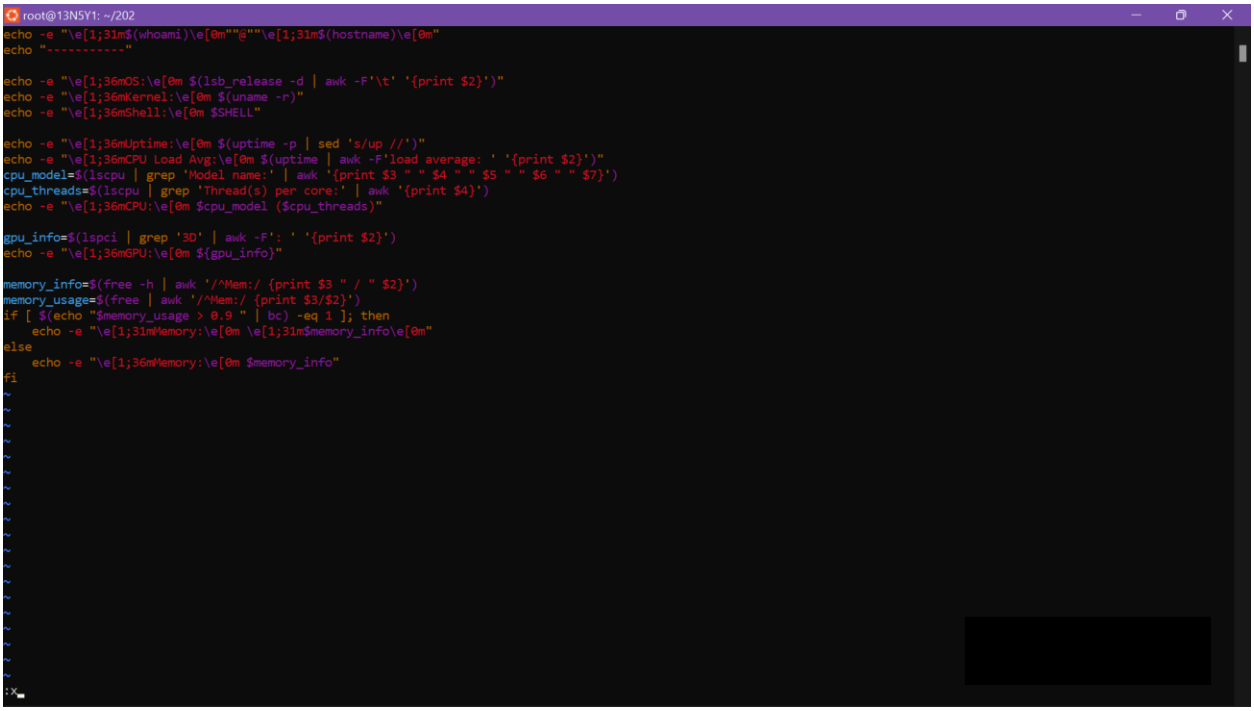
The above picture shows implementation of my script followed by Neofetch's output.

Line by line:

1. Make a directory named "202" in the current directory.
2. Change directory to 202
3. Create script.sh file and open it in Vim editor

4. Open script.sh in Vim
5. Make script.sh executable
6. Run script
7. Run Neofetch

Code:



Linux commands used:

echo -e: Allows escape commands to be used with echo. Used to give color to output here. Everything between \e and m is the escape command e.g. [1;36 is cyan color, [0 is default (white).

whoami: prints username.

hostname: prints hostname / name of device.

lsb_release -d: print OS details.

awk -F '\t': extracts text after each tab space, {print $2}: prints second word found.

```
root@13N5Y1:~/202# lsb_release -d
No LSB modules are available.
Description:    Ubuntu 24.04.1 LTS
```

The image shows regular output of lsb_release -d. The tab(\t) is between Description: and Ubuntu, awk -F '\t' separates the everything before and after the tab and {print $2} selects the second part (Ubuntu...)

uname -r: prints kernel release information.

$SHELL: The environment variable that stores shell's absolute path (/bin/zsh).

uptime -p: Prints how long device has been up and running.

sed: Used for editing strings. In this case it searches for every instance of 'up' (s/ up), and replaces it with an empty string (//).

```
root@13N5Y1:~/202# uptime -p
up 3 hours, 46 minutes
```

Default output of uptime -p.

Awk -F "load average: " splits the output of uptime into two parts. One before (and including) load average and the other everything after it. {print $2} prints the second part of the spliced string.

lscpu: prints CPU information.

grep: Finds select string from output.

lspci: List PCI devices (network card, GPU) information.

memory_info: user defined variable.

if: if statement to give output if certain conditions are met

bc: inbuilt calculator tool to calculate if memory usage is greater than 90 percent

else: else condition if true condition is false

fi: end if statement

```
nc -z 1.1.1.1 53 > /null 2>&1 && echo -e "\e[1;32mInternet ON\e[0m" || echo "\e[1;31mDevice not connected to internet\e[0m"
```

Added this to the end of the script to check network connection using netcat to scan for open ports on Cloudflare's DNS server. -z option does not send any data. It follows by sending the standard output and error of nc to /null (should've been /dev/null according to standards) so it does not print in terminal.