Module 3: Scripting with Linux

James Way & Venetia Furtado

Date: 11/17/2024

Aim: Creation and running an executable script in Linux to print useful information about the Linux Kernel.

1. Create an executable script

A bash script was written to perform the following functions:

i. Display the current user and terminal.
ii. Display the current running processes.
iii. Display the current data and time, kernel version.
iv. Display the kernel dump.

```
#!/bin/bash

echo "---- System Information Script ----"

# Display the current user and terminal
echo "Current User: $(whoami)"
echo "Terminal: $TERM"
echo "Shell: $SHELL"

# Display running processes
echo -e "\nCurrent Running Processes:"
ps aux

# Display current date, time, and kernel version
echo -e "\nCurrent Date and Time:"
date
echo -e "\nKernel Version:"
uname -r

# Display kernel dump
echo -e "\nKernel Dump (dmesg output):"
dmesg | tail -n 20   # Limiting output to the last 20 lines for brevity

echo -e "\n---- End of System Information ----"
```

Figure 1: Snapshot of the bash script.

The command used to make the bash script executable was: chmod +x terminalInfo.sh

The outputs have been shown in Figure 2 below.

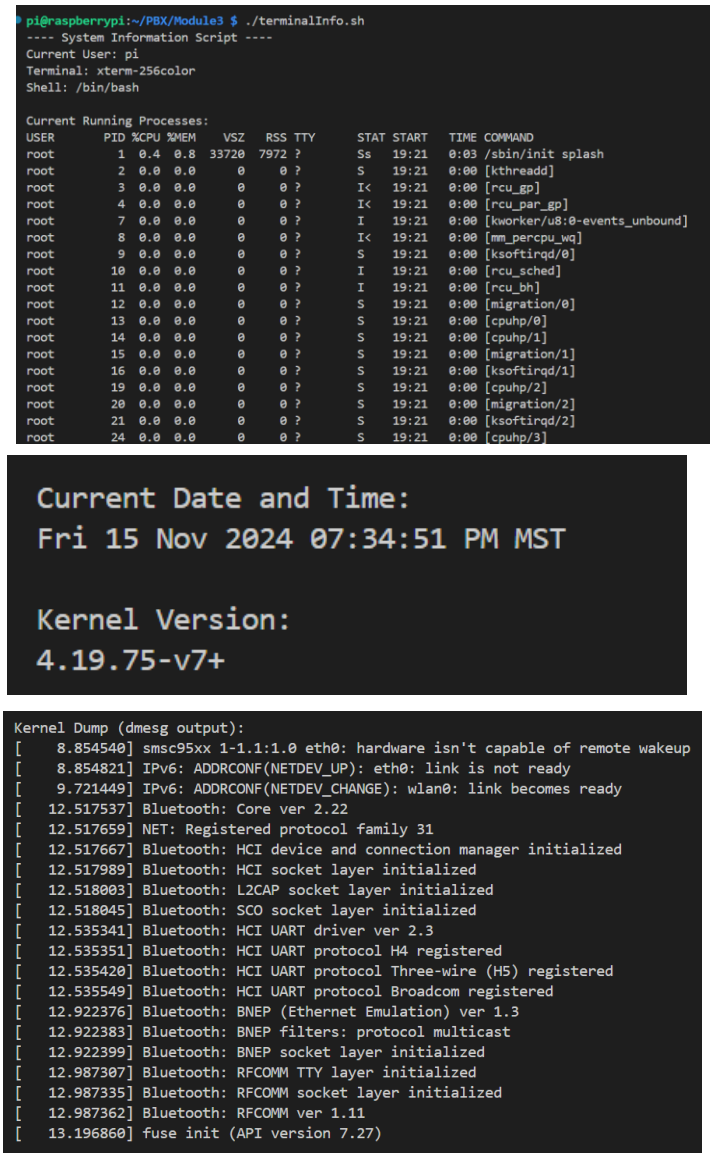

Figure 2: Output of the bash script.

2. **Make the bash script run in boot time**

   To make the script run in boot time, Raspberry Pi's system (the default init system) was used to create a service file that gets executed upon boot.

   The systemd service was created: *sudo nano /etc/systemd/system/terminalInfo.service*

   To enable the service to run at boot: *sudo systemctl enable terminalInfo.service*

   To start the service run: *sudo systemctl start terminalInfo.service*

   To check the status of the service: *sudo systemctl status terminalInfo.service* which shows "active".

   To reboot system: *sudo reboot now*

To get the syslog run: *sudo cat /var/log/syslog*
The syslog shows that the script runs after the Raspberry Pi is reconnected with the network as specified in the service file.

A copy of the syslog file has been submitted for review.

```
[Unit]
Description=Run terminalInfo at boot
After=network.target

[Service]
ExecStart=/home/pi/PBX/Module3/terminalInfo.sh
Type=oneshot
RemainAfterExit=true

[Install]
WantedBy=multi-user.target
```

Figure 3: Systemd service file.

References:

[1] BASH Programming - Introduction HOW-TO

[2] https://linuxconfig.org/how-to-autostart-bash-script-on-startup-on-raspberry-pi