PROJECT 2

MODULE 3

Creating an executable script:

An executable Python script for creating and running the task to print the useful information about Linux Kernel has been completed. The list of functions implemented and their description is given below.

- Listing all running processes: The command ‘ps aux’ is used to list all currently running processes along with detailed information such as the user, process ID, CPU, memory usage, start time, and command that started the process.
- Printing kernel name: The command ‘uname –kernel-name’ is used to print the kernel's name.
- Printing Kernel version: The command ‘uname –kernel-version’ is used to print the version of the kernel.
- Kernel dump: The command ‘dmesg’ is used to display the kernel ring buffer messages, which include boot and system-related messages that are used for debugging.
- Printing device IP address: The command ‘ifcong | grep ‘broadcast’ | awk ‘{print $2}’’ is used to display the IP address of the device. ‘ifconfig’ is to display the network interface configuration. ‘grep ‘broadcast’ is used to filter the output to get the lines containing ‘broadcast’. ‘awk ‘{print $2}’’ is used to extract the second field which is the IP address of the device.
- Listing background processes: The command ‘ps aux | grep “^[^]* s”’ is used to display the processes that are in Sleeping Status(s).
- Listing the number of processes per user: The command ‘ps -e -o user= | sort | uniq -c’ is used to list the processes by the user, it sorts them and counts the number of processes per user.
- Real-time Linux monitoring: The command ‘top -b -n 1’ is used to monitor the real-time Linux. ‘top’ is run in batch mode ‘-b’ to do the scripting. ‘-n 1’ is used to specify that only one iteration of the output should be displayed.
- Display Disk utilization: the command ‘df -h’ is used to display the disk usage for all mounted filesystems in human-readable format.
- Display memory consumption: The command ‘free -m’ is used to show the memory usage specifying the total, used, free, and available memory.
- Display RAM statistics: The command ‘lscpu’ is used to display detailed information about the CPU architecture, specifying the number of CPUs, cores, threads, and more.
- Display the physical memory statistics: The command ‘ free -m | grep 'Mem' | awk '{print \"Physical Memory Stats in MB: \", \"Total:\", $2, \"Used:\", $3,\"Free:\", $4, \"Shared:\", $5,\"Buff/Cache:\", $6, \"Available:\", $7}’’ is used to extract the physical memory statistics from memory consumption stats. It gives the breakdown of total, used, free, shared, buffered/cache, and available memory.

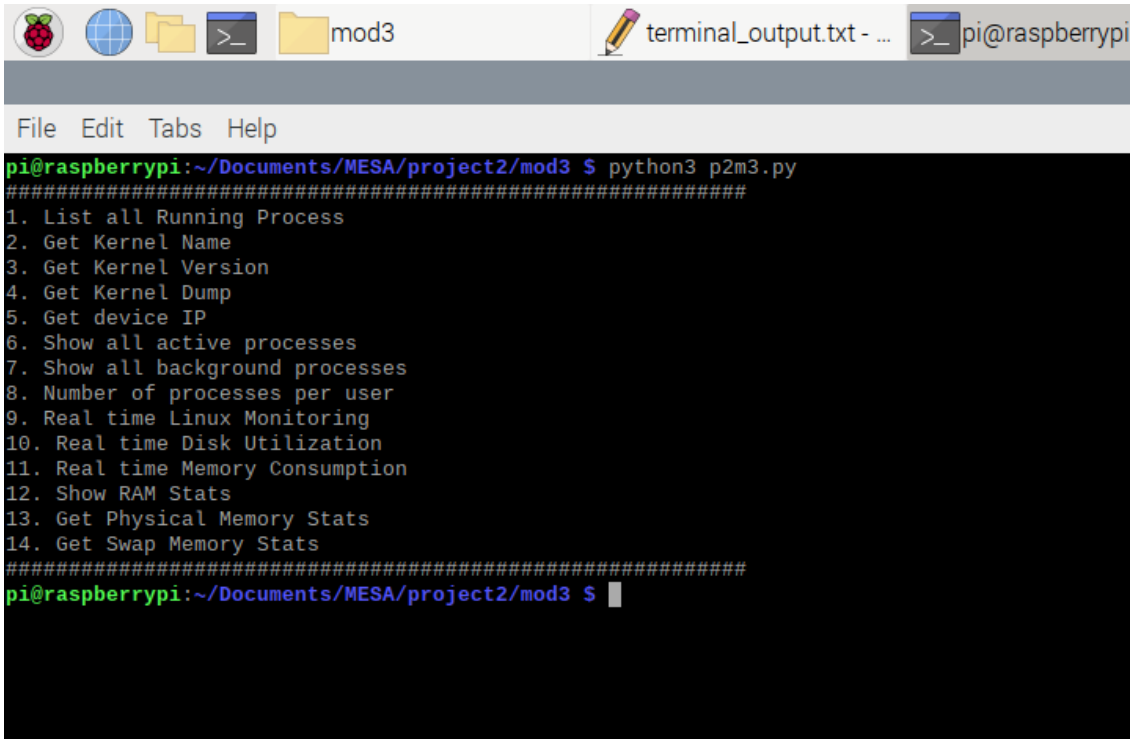
- Display Swap memory stats: like the above physical memory stats, we get the swap memory stats from memory consumption by applying filters.

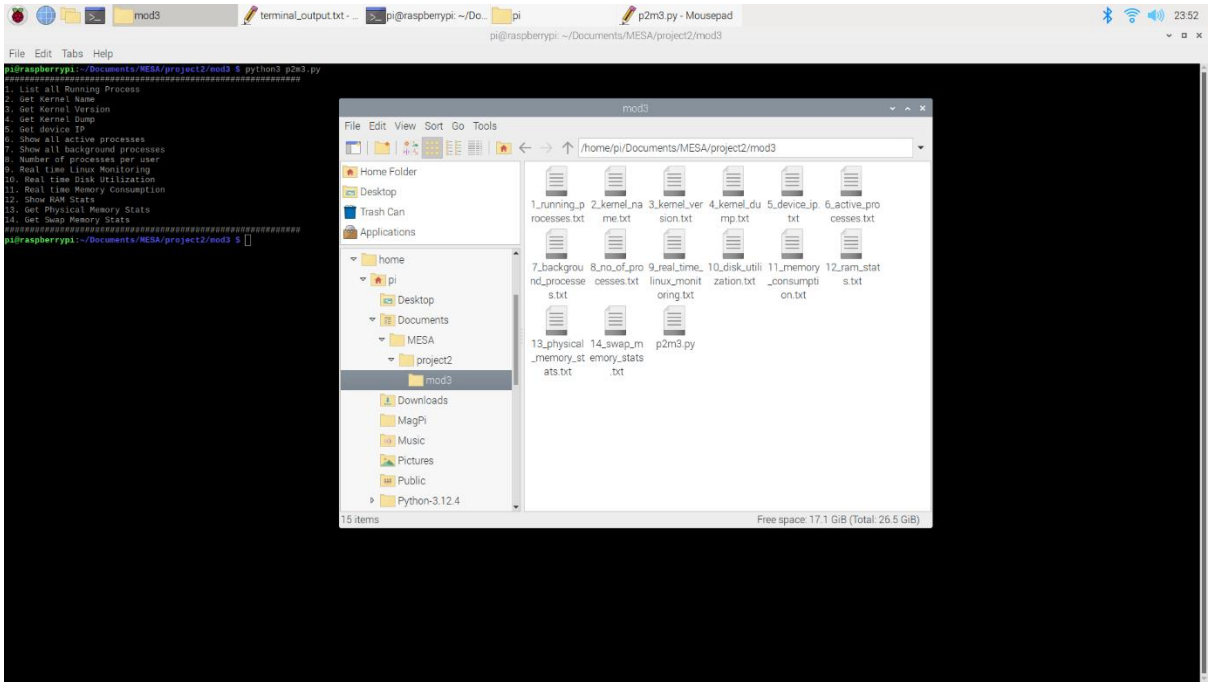
Bonus section:

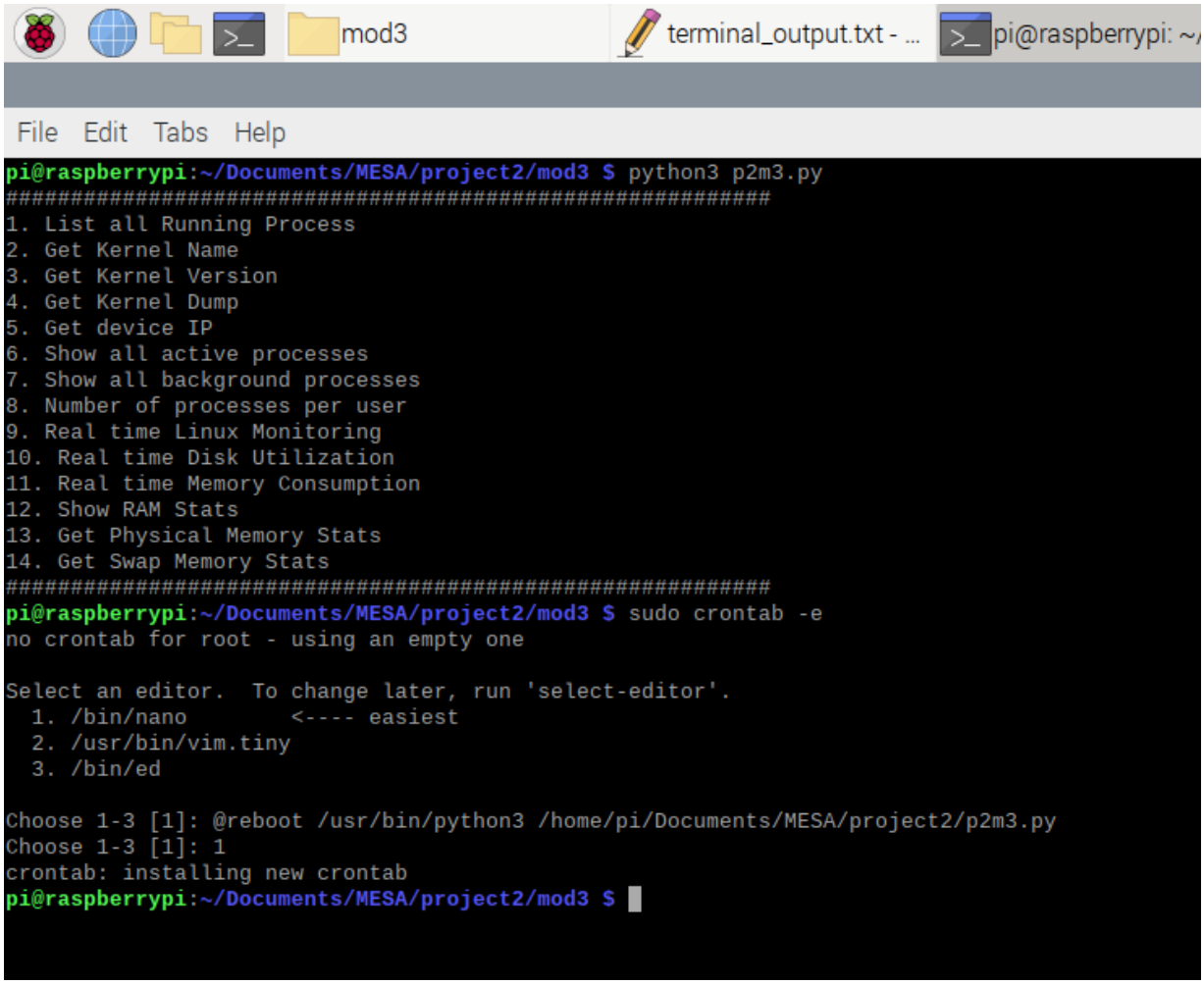
We have configured the Python script to run at the system start-up using crontab. Executing the command ‘sudo crontab -e’ opens the editor. We then entered ‘1’ to choose the nano editor. A file opened and we added the line ‘@reboot /usr/bin/python3 /home/pi/Documents/MESA/project2/mod3/p2m3.py’. Saved the file and closed it.

This will ensure the Python code is executed during bootup.

Screenshots:




mod3
terminal_output.txt - ...
pi@raspberrypi: ~/Do...
pi
p2m3.py - Mousepad
23:52
pi@raspberrypi: ~/Documents/MESA/project2/mod3
File Edit Tabs Help
pi@raspberrypi:~/Documents/MESA/project2/mod3 $ python3 p2m3.py
1. List all Running Process
2. Get Kernel Name
3. Get Kernel Version
4. Get Kernel Dump
5. Get device IP
6. Show all active processes
7. Show all background processes
8. Number of processes per user
9. Real time Linux Monitoring
10. Real time Disk Utilization
11. Real time Memory Consumption
12. Show RAM Stats
13. Get Physical Memory Stats
14. Get Swap Memory Stats
pi@raspberrypi:~/Documents/MESA/project2/mod3 $
mod3
File Edit View Sort Go Tools
/home/pi/Documents/MESA/project2/mod3
Home Folder
Desktop
Trash Can
Applications
home
pi
Desktop
Documents
MESA
project2
mod3
Downloads
MagPi
Music
Pictures
Public
Python-3.12.4
1_running_processes.txt
2_kernel_name.txt
3_kernel_version.txt
4_kernel_dump.txt
5_device_ip.txt
6_active_processes.txt
7_background_processes.txt
8_no_of_processes.txt
9_real_time_linux_monitoring.txt
10_disk_utilization.txt
11_memory_consumption.txt
12_ram_stats.txt
13_physical_memory_stats.txt
14_swap_memory_stats.txt
p2m3.py
15 items
Free space: 17.1 GiB (Total: 26.5 GiB)


mod3
terminal_output.txt - ...
pi@raspberrypi: ~/
File Edit Tabs Help
pi@raspberrypi:~/Documents/MESA/project2/mod3 $ python3 p2m3.py
##
1. List all Running Process
2. Get Kernel Name
3. Get Kernel Version
4. Get Kernel Dump
5. Get device IP
6. Show all active processes
7. Show all background processes
8. Number of processes per user
9. Real time Linux Monitoring
10. Real time Disk Utilization
11. Real time Memory Consumption
12. Show RAM Stats
13. Get Physical Memory Stats
14. Get Swap Memory Stats
##
pi@raspberrypi:~/Documents/MESA/project2/mod3 $ sudo crontab -e
no crontab for root - using an empty one
Select an editor. To change later, run 'select-editor'.
1. /bin/nano <---- easiest
2. /usr/bin/vim.tiny
3. /bin/ed
Choose 1-3 [1]: @reboot /usr/bin/python3 /home/pi/Documents/MESA/project2/p2m3.py
Choose 1-3 [1]: 1
crontab: installing new crontab
pi@raspberrypi:~/Documents/MESA/project2/mod3 $

mod3 terminal_output.txt - ... pi@raspberrypi: ~

File Edit Tabs Help

```
  GNU nano 3.2

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
@reboot /usr/bin/python3 /home/pi/Documents/MESA/project2/mod3/p2m3.py
```