PYTHON FUNDAMENTALS
(PYTHON OS INFO)

Yap Ching Siong
CFC2407

Contents

Overview of the script

This script is created to automate the display of operating system information.

After executing the script, the following information shall be printed on the terminal:

1. OS Version.
2. Private IP address, Public IP address Default Gateway.
3. Hard disk size, free and used space.
4. Top five directories and their size (the directories involved in calculation depends on the path user executes this script)
5. CPU usage which refreshes every 10 seconds.

The os.system() method executes command in subshell of the operating system it is executed on.

The command is written as string which is pass into os.system() method as an argument for execution.

Example: *date* command can be used on Linux system to display current date, time, year.

To use *date* command for Linux system through os.system() method in python3 environment, we'll just need to execute os.system("date") in python3 environment.

For this project, Linux operating system will be used.

Detail code explanation will be in the following sections.

OS version

The following codes will import os module and print out the system information on the terminal.

```
import os

print("OS version: ")
os.system("uname -a")
print("\n")
os.system("sleep 2")
```

import the os module so that os function and os.system() method can be use in our script.

import os

Print the sentence “OS version:” to inform user that the following sentence will contain OS version
information.

print("OS version: ")

uname command to print system information

-a flag to print all, which includes kernel name, network node hostname, kernel release, kernel
version, machine hardware name, operating system.

os.system("uname -a")

print \n to insert a newline for separation with the next line for tidy output.

sleep 2, pause for 2 seconds before executing the next command, for user to look at the terminal
output.

print("\n")

os.system("sleep 2")

```
┌──(ycs㉿kali)-[~]
└─$ python3 PythonOSInfo.py
OS version:
Linux kali 5.15.0-kali3-amd64 #1 SMP Debian 5.15.15-2kali1 (2022-01-31) x86_64 GNU/Linux
```

IP Address and Default Gateway

The following codes will print out the private IP address, public IP address and default gateway of use by the machine.

```
print("Private IP Address: ")
os.system("ifconfig | grep broadcast | awk '{print $2}'")
print("\n")
os.system("sleep 2")

print("Public IP Address: ")
os.system("curl ifconfig.io")
print("\n")
os.system("sleep 2")

print("Default Gateway: ")
os.system("route | grep UG | awk '{print $2}' | uniq")
print("\n")
os.system("sleep 2")
```

Notify user the following line is information for the private IP address.

print("Private IP Address: ")

ifconfig to return the view the network configuration of the network interface on the system,
which also includes the private ip address.

grep broadcast, to display the line with the broadcast keyword, the line contains the private ip
address.

awk '{print $2}' to print out the second string of the line which is our private ip address.

os.system("ifconfig | grep broadcast | awk '{print $2}'")

print \n to insert a newline for separation with the next line for tidy output.

sleep 2, pause for 2 seconds before executing the next command, for user to look at the terminal
output.

print("\n")

os.system("sleep 2")

```
Private IP Address:
192.168.23.128
```

```
# Notify user the following line is information for the public IP address.
print("Public IP Address: ")

# curl is a command line to transfer data, in this case retrieve data from 'ifconfig.io' which is our
#  public ip address.
os.system("curl ifconfig.io")

# print \n to insert a newline for separation with the next line for tidy output.
# sleep 2, pause for 2 seconds before executing the next command, for user to look at the terminal
#  output.
print("\n")
os.system("sleep 2")
```

```
Public IP Address:
103.6.150.242
```

```
# Notify user the following line is information for the default gateway.
print("Default Gateway: ")

# ifconfig to return the view the network configuration of the network interface on the system,
#  which also includes the private ip address.
# grep UG, to display the line with the UG keyword, the line contains the default gateway.
# awk '{print $2}' to print out the second string of the line which is our default gateway.
# uniq to display only one of the duplicated default gateway.
os.system("route | grep UG | awk '{print $2}' | uniq")

# print \n to insert a newline for separation with the next line for tidy output.
# sleep 2, pause for 2 seconds before executing the next command, for user to look at the terminal
#  output.
print("\n")
os.system("sleep 2")
```

```
Default Gateway:
192.168.23.2
```

Hard disk size

The following codes will display the used and free physical hard disk size.

```
print("Hard Disk Size: ")
os.system("df -h")
print("\n")
os.system("sleep 2")
```

Notify user the following line is information for the hard disk size.

print("Hard disk size: ")

df command is use to display amount of space taken up by different drives.

-h flag to print the size in human readable format, eg. M(Megabytes), G(Gigabytes).

os.system("df -h")

print \n to insert a newline for separation with the next line for tidy output.

sleep 2, pause for 2 seconds before executing the next command, for user to look at the terminal
output.

print("\n")

os.system("sleep 2")

```
Hard Disk Size:
Filesystem      Size  Used Avail Use% Mounted on
udev            944M     0  944M   0% /dev
tmpfs           198M  1.3M  197M   1% /run
/dev/sda1        78G   13G   61G  18% /
tmpfs           987M     0  987M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           198M   96K  198M   1% /run/user/1001
```

udev is the virtual directory for the /dev directory.

/dev/sda1 is the physical hard drive.

tmpfs are used by Linux processes as temporary filesystems for running the operating system.

Top five directories and their size

The following codes will calculate the size of directories and display the top five directories with the largest file size. The directories in consideration is determined by the path where is script is located.

```
print("Top Five Largest Directories And Their Size: ")
os.system("du -h | sort -h | tail -n 5")
print("\n")
os.system("sleep 2")
```

Notify user the following line is information on the top five directories and their size.

print("Top Five Largest directories and their size: ")

du command estimates the file size usage

-h flag to print size in human readable format eg. M(Megabytes), G(Gigabytes).

sort -h, is to sort by human readable numbers eg. by M(Megabytes), G(Gigabytes).

tail -n 5, to capture the bottom 5 information displayed.

os.system("du -h | sort -h | tail -n 5")

print \n to insert a newline for separation with the next line for tidy output.

sleep 2, pause for 2 seconds before executing the next command, for user to look at the terminal
output.

print("\n")

os.system("sleep 2")

```
Top Five Largest Directories And Their Size:
130M    ./.cache/mozilla
130M    ./.cache/mozilla/firefox
130M    ./.cache/mozilla/firefox/pf62h1jv.default-esr
138M    ./.cache
457M    .
```

CPU Usage

The following codes display CPU usage which will be refreshing every 10 seconds.

```
print("CPU Usage (Ctrl + C to Exit): ")
os.system("top -d 10")
```

Notify user the following line is information on the CPU usage and user should use ctrl + c to exit
from the screen.

print("CPU Usage (Ctrl + C To Exit): ")

top command to display linux processes, which also displays the CPU usage information.

-d 10, flag is to set the refresh rate to 10 seconds.

os.system("top -d 10")

```
CPU Usage (Ctrl + C to Exit):
top - 09:44:26 up 2 days,  5:17,  1 user,  load average: 0.24, 0.21, 0.19
Tasks: 215 total,   1 running, 214 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.1 us,  0.2 sy,  0.0 ni, 99.5 id,  0.1 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1973.4 total,    108.7 free,    681.2 used,   1183.5 buff/cache
MiB Swap:    975.0 total,    717.3 free,    257.7 used.   1081.6 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    621 root      20   0  654892 234912  64228 S   1.0  11.6  39:15.89 Xorg
    875 ycs       20   0 1226028  64976  51204 S   1.0   3.2  10:57.96 xfwm4
    487 root      20   0  313432   6996   6260 S   0.5   0.3   8:33.10 vmtoolsd
 792675 ycs       20   0  426156 102244  79104 S   0.5   5.1   0:01.93 qterminal
      1 root      20   0  170772  10468   7516 S   0.0   0.5   0:09.23 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.10 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
      9 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
     10 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
     11 root      20   0       0      0      0 S   0.0   0.0   0:04.56 ksoftirqd/0
     12 root      20   0       0      0      0 I   0.0   0.0   1:13.12 rcu_sched
     13 root      rt   0       0      0      0 S   0.0   0.0   0:00.58 migration/0
     15 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     16 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     17 root      rt   0       0      0      0 S   0.0   0.0   0:00.54 migration/1
     18 root      20   0       0      0      0 S   0.0   0.0   0:00.96 ksoftirqd/1
     20 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/1:0H-events_highpri
     21 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/2
     22 root      rt   0       0      0      0 S   0.0   0.0   0:00.59 migration/2
     23 root      20   0       0      0      0 S   0.0   0.0   0:01.69 ksoftirqd/2
     25 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/2:0H-events_highpri
```

Reference

Python | os.system() method

https://www.geeksforgeeks.org/python-os-system-method/

How to check OS and version using a Linux command [duplicate]

https://unix.stackexchange.com/questions/88644/how-to-check-os-and-version-using-a-linux-command

How to Check Disk Space in Linux

https://phoenixnap.com/kb/linux-check-disk-space

How to Check CPU Utilization in Linux with Command Line

https://phoenixnap.com/kb/check-cpu-usage-load-linux

How To Find Largest Top 10 Files and Directories On Linux / UNIX / BSD

https://www.cyberciti.biz/faq/how-do-i-find-the-largest-filesdirectories-on-a-linuxunixbsd-filesystem/

End