# LAB – WRITE INFORMATION TO A FILE

## OBJECTIVE

In this lab you will write an application that reads device information from a file, strips off excess whitespace characters, and displays the information using formatting options that are provided in Python, to create attractive output.

The application will then create a string containing all the device information; write out the line of device information to the file; then read in the file and display the line of device info.

You will be using the file **devices-01.txt**, which includes the following information, each on its own line:

- Device name
- IP address
- OS type
- Username
- Password

## PART 1

Open a Linux terminal and switch to the lab directory

### STEP 1: OPEN A TERMINAL WINDOW

Double-click the Terminal icon on the desktop to open the terminal window for use in this lab.

### STEP 2: CHANGE DIRECTORY

Change to the directory **labs/prne/** in the user home directory, which holds the files for the course labs.

```
~$ cd labs/prne/
```

## PART 2

Open **Visual Studio Code**, create a new file and save it with a filename of **write-information-to-a-file.py**. Ensuring to save the file in the **~/labs/prne/** directory, as otherwise the code will require modification to find the associated files that are used.

### STEP 1: OPEN FILE FOR READING

The first thing the application should do is open the file **devices-01.txt** for reading.

```
# Open file for reading
file = open('devices-01.txt', 'r')
```

## STEP 2: READ AND STRIP WHITE SPACE

Read in the information from the file, one line at a time and strip off extra whitespace characters for each line.

```
# Read the file one line at a time and strip off any extra whitespace
# characters for each line
name = file.readline().strip()
ip_address = file.readline().strip()
os_type = file.readline().strip()
username = file.readline().strip()
password = file.readline().strip()
```

## STEP 3: DISPLAY THE DEVICE INFORMATION

Display the device information that has been read in, using the print and string formatting capabilities of Python. Display the information as a table of devices (although there will only be one device), with columns for the fields of name, IP address, OS type, username, and password.

```
# Display the information using string formatting
print('--- Device info nicely formatted -------------------------------')
print('')
print('Name     IP address      OS       Username   Password')
print('-----    --------------- -----    --------   --------')
print('{0:8} {1:15} {2:8} {3:10} {4:10}'.format(name, ip_address,
                                                os_type, username,
                                                password))

print('')
print('------------------------------------------------------------------')
```

## STEP 4: CREATE A STRING

Create a string containing all the device information that is separated by commas.

```
# Create comma-separated string of device information attributes
device_info = name  # start with device name
device_info = device_info + ',' + ip_address  # add comma and IP address
device_info = device_info + ',' + os_type     # add comma and os-type
device_info = device_info + ',' + username    # add comma and username
device_info = device_info + ',' + password    # add comma and password
```

## STEP 5: WRITE INFORMATION TO A FILE

Write the string to a file, displaying status.

```
# Write device information string to file
print('')
print('--- Writing device information to file -------------------------')
print('')
outfile = open('devices-01-out.csv', 'w')  # open the output file
outfile.write(device_info)  # write the line of device information
outfile.write('\n')         # with 'write' we must add ending newline char
outfile.close()             # close file when writing is complete
print('--- Device information written to file -------------------------')
print('')
```

## STEP 6: READ THE INFORMATION FROM THE FILE

Read in the device information from the file

```
# Open the file that was just created
infile = open('devices-01-out.csv', 'r')  # open the new one-line file
device_info = infile.readline().strip()   # read the line from the file
infile.close()  # close the file
```

## STEP 7: DISPLAY FILE CONTENT ON THE SCREEN

Display in the terminal the contents of the file written.

```
# Display the information from the file that was just created
print('')
print('--- Device info read from file we wrote ------------------------')
print('')
print('Device Info: ', device_info)
print('')
print('----------------------------------------------------------------')
print('')
```

## STEP 8: SAVE AND RUN THE APPLICATION

Save you application and then run it from the terminal rather than from within visual studio code.

```
~/labs/prne$ python3 write-information-from-a-file.py
```

Verify the output in the terminal from your application is comparable to below.

```
devasc@labvm:~/labs/prne$ python3 write-information-to-a-file.py
--- Device info nicely formatted -------------------------------

Name     IP address      OS       Username   Password
-------- --------------- -------- ---------- ----------
ios-01   10.30.30.1      ios      admin      cisco

----------------------------------------------------------------

--- Writing device information to file -----------------------

--- Device information written to file -----------------------


--- Device info read from file we wrote ----------------------

Device Info:  ios-01,10.30.30.1,ios,admin,cisco

----------------------------------------------------------------
```

## STEP 9: VERIFY APPLICATION HAS CREATED THE OUTPUT FILE

In the terminal window, check that the file **devices-01-out.csv** has been created by using the **ls** command.

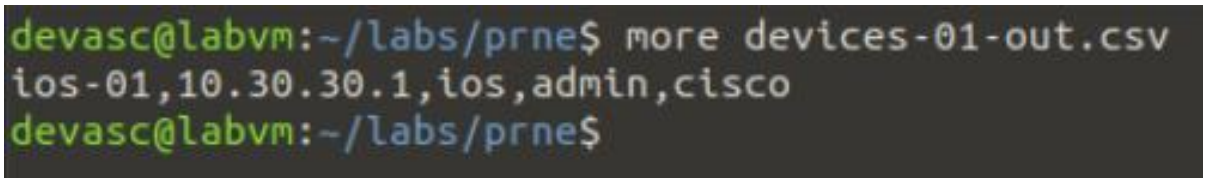
```
~/labs/prne$ ls
```

```
devasc@labvm:~/labs/prne$ ls
devices-01-out.csv  devices-07.txt  devices-14.json
devices-01.txt      devices-08.txt  hello-device.py
devices-02.csv      devices-09.txt  ip-routes.txt
devices-03.txt      devices-10.txt  read-information-from-a-file.py
devices-04.txt      devices-11.txt  write-information-to-a-file.py
devices-05.txt      devices-12.txt
devices-06.txt      devices-13.txt
devasc@labvm:~/labs/prne$
```

### STEP 10: VERIFY THE CONTENT OF THE OUTPUT FILE

In the terminal window, check the content of the file **devices-01-out.csv** by using the **more** command to ensure that device information attributes have been added in csv format.

```
~/labs/prne$ more devices-01-out.csv
```




## PART 3 (OPTIONAL BUT HIGHLY RECOMMENDED)

As this lab is completed in NETLAB+ and your code files will be erased when the reservation ends, it is advisable to save your files in GitHub under your repository for this course.