TopGear Assignments

Python Programming – L2

Case study Projects

Author: Jayapathi Ramamohan

jayapathi.ramamohan@wipro.com

Estimated Effort: 6PDs

Total Points: 600

Required VDI: Linux

Date: 9-March-2018

This set of case studies require application of following concepts of Python.

- File operations
- Regular expressions
- Data structures
- Multithreading
- Thread queues

To be considered as correct submission, code should meet the following guidelines:

- Source code filename : as specified in each exercise
- Input filename & output filename should not be hardcoded in program, unless it is stated as part of the problem specification
- Output should be as per the sample shown in each exercise
- Individual files have to be uploaded into gitlab, not as a single zip file

Case study– 1:   Filename: cs1.py

Implement a multithreaded solution that takes file name(s) as command-line argument and prints the count of lines, words and characters for each line.

Spaces, tabs and newline characters are considered as word delimiters.

Each file should be analyzed by one thread. The thread must get the count of lines & words for the file, place the results in a queue and should terminates.

The primary thread should read the queue and produce output sorted on filename.

Sample program run and output is shown below.

```
$  cs1.py  x.dat  a.txt  b.dat  k.txt  g.txt
a.txt        123        347
b.dat       1403        712
g.txt     110897     256104
k.txt       1042       8217
x.dat       4571      10392
```

Case study– 2:   Filename: cs2.py

Implement a python program that analyzes python source files in a given directory and produces class dependency list, along with the source filename in which the class definition is found, in the following format.

```
Class name1 [filename]
    Derived Classname1 [filename]
    Derived Classname2 [filename]
    Derived Classname3 [filename]
    …
Class name2 [filename]
    Derived ClassnameA [filename]
    Derived ClassnameB[filename]
    Derived ClassnameC [filename]
    …
…
```

For example, running the program cs2.py on the directory **/usr/lib/python2.6/site-packages/virtinst,** resulted in the output that is partially shown

$ python cs2.py /usr/lib/python2.6/site-packages/virtinst

some output not shown here

```
Distro [DistroInstaller.py]
   DebianDistro [OSDistro.py]
   GenericDistro [OSDistro.py]
   MandrivaDistro [OSDistro.py]
   NetWareDistro [OSDistro.py]
   RedHatDistro [OSDistro.py]
   SunDistro [OSDistro.py]
   SuseDistro [OSDistro.py]
Guest [CapabilitiesParser.py]
   FullVirtGuest [FullVirtGuest.py]
   ParaVirtGuest [ParaVirtGuest.py]
```

```
ImageFetcher [ImageFetcher.py]
   LocalImageFetcher [ImageFetcher.py]
   URIImageFetcher [ImageFetcher.py]
Installer [Installer.py]
   DistroInstaller [DistroInstaller.py]
   ImageInstaller [ImageInstaller.py]
   ImportInstaller [ImportInstaller.py]
   LiveCDInstaller [LiveCDInstaller.py]
   PXEInstaller [PXEInstaller.py]
Interface [Interface.py]
   _InterfaceCompound [Interface.py]
   InterfaceEthernet [Interface.py]
   ...
   InterfaceProtocolIPv6 [Interface.py]
   InterfaceVLAN [Interface.py]
```

some output not shown here

```
NodeDevice [NodeDeviceParser.py]
   PCIDevice [NodeDeviceParser.py]
   SCSIBus [NodeDeviceParser.py]
   ...
   SystemDevice [NodeDeviceParser.py]
   USBBus [NodeDeviceParser.py]
   USBDevice [NodeDeviceParser.py]
```

some output not shown here

```
VirtualDevice [VirtualDevice.py]
   VirtualAudio [VirtualAudio.py]
   VirtualCharDevice [VirtualCharDevice.py]
   VirtualController [VirtualController.py]
   ...
   VirtualNetworkInterface [VirtualNetworkInterface.py]
   VirtualVideoDevice [VirtualVideoDevice.py]
   VirtualWatchdog [VirtualWatchdog.py]
```

Note: You can create a set of dummy python program files that have classes (both base class & derived classes defined), which can be used to test your code for correctness.

--- END ---