Exercise 7 – Data Storage and File Handling

Objective

To use some of the Python 3 file handling methods, as well as the pickle and gzip modules.

Questions

1. Write a Python script to list all the unused port numbers in the /etc/services file between 1 and 200

    Steps:

    Become familiar with the input file - view it first

    Write the main code to read the services file one line at a time

    Use string functions or a regular expression to:

    Ignore lines starting with a # comment character

    Ignore lines that just consist of "white-space"

    The /etc/services has several columns separated by white-space

    – Use split or a regular expression to isolate the port/protocol field

    – Use another split or regular expression to isolate the port number

    – Don't forget to stop at port number 200!

    – Note that many port numbers have > 1 entry

    On Windows the file is in 'C:\WINDOWS\system32\drivers\etc\services' or in 'C:\WINNT\system32\drivers\etc\services'.

    On OSX the file has unused ports marked as 'Unassigned'. Therefore, we have an addition requirement: ignore all lines that start with the comment delimiter '#'.

    Many port numbers have more than one entry in the file, but you may assume they are in order.

    Hints: Open the file.
    Read the file line-by-line using a for loop.
    Consider using a set or a dictionary to hold the port numbers.
    Be careful of comparing strings and int - you will have to convert the port number to an integer.

2. Using the data in **country.txt**, construct a Python dictionary where the country name is the key and the other record details are stored in a list as the value. Store (pickle) this dictionary into a file named 'country.p'.

   Notice the size of the file compared to the original, and then change the program to use gzip.

3. Now write a program which reads the pickled dictionary and displays it onto the console.

   If time allows, convert your pickle to use a shelve.

Solutions

Question 1

This solution uses regular expressions and sets. A common mistake with this approach is to forget to convert the captured port number to an int, required since range returns an integer.

```
import sys
import re

if sys.platform == 'win32':
    file = r'C:\WINDOWS\system32\drivers\etc\services'
else:
    file = '/etc/services'

ports = set()

for line in open(file, 'r'):
    m = re.search(r'(\d+)/(udp|tcp)', line)
    if m:
        port = int(m.group(1)) # Or m.groups()[0]
        if port > 200: break
        ports.add(port)

# Subtract used port numbers from full set of ports
print(set(range(1, 201)) - ports)
```

Questions 2 & 3

```python
import pickle
import gzip
import shelve

# Using a compressed pickle.
country_dict = {}
for line in open('country.txt', 'r'):
    name, *row = line.split(',')
    country_dict[name] = row

outp = gzip.open('country.p', 'wb')
pickle.dump(country_dict, outp)
outp.close()

# Using a shelve.
db = shelve.open('country')
for country in country_dict.keys():
    db[country] = country_dict[country]

db.close()
db = shelve.open('country')
print(db['Belgium'])
db.close()
```