CNSM4: Distributed and Cloud Computing

Practical 1.01: Printing your machine's name and IPv4 address

John Jennings

LIT

Introduction and Overview

This practical aims to kick-start your development using Python. We wil be using Python's core networking library, **socket**, through some simple examples. Python's socket module has both class-based and instances-based utilities. The difference between a class-based and instance-based method is that the former doesn't need an instance of a socket object. This is a very intuitive approach. For example, in order to print your machine's IP address, you don't need a socket object. Instead, you can just call the socket's class-based methods. On the other hand, if you need to send some data to a server application, it is more intuitive that you create a socket object to perform that explicit operation.

Printing your machine's name and IPv4 address

Sometimes, you need to quickly discover some information about your machine, for example, the host name, IP address, number of network interfaces, and so on. This is very easy to achieve using Python scripts.

1. Get Set Up

You need to install Python on your machine before you start coding. Python comes preinstalled in most of the Linux distributions. For Microsoft Windows operating system, you can download binaries from the Python website: http://www.python.org/download/

You may consult the documentation of your OS to check and review your Python setup. After installing Python on your machine, you can try opening the Python interpreter from the command line by typing python. This will show the interpreter prompt, >>>, which should be similar to the following output:

```
~$ python  Python 2.7.1+ (r271:86832, Apr 11 2011, 18:05:24)  [GCC 4.5.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

2. The Code

Lets try this in the Python interpreter interactively.

- First, we need to import the Python socket library with the following command:

```
>>> import socket
```

- Then, we call the gethostname() method from the socket library and store the result in a variable as follows:

```
>>> host_name = socket.gethostname()
>>> print ("Host name: %s" %host_name)
Host name: debian6
>>> print ("IP address: %s" %socket.gethostbyname(host_name) )
IP address: 127.0.1.1
```

- The entire activity can be wrapped in a free-standing function, `print_machine_info()`, which uses the built-in socket class methods.
- We call our function from the usual Python `__main__` block. During runtime, Python assigns values to some internal variables such as `__name__`. In this case, `__name__` refers to the name of the calling process. When running this script from the command line, as shown in the following command, the name will be `__main__`, but it will be different if the module is imported from another script. This means that when the module is called from the command line, it will automatically run our `print_machine_info` function; however, when imported separately, the user will need to explicitly call the function.
- In a text editor of your choice create a new file named `1_1_local_machine_info.py` and add the following code:

```
#!/usr/bin/env python

import socket

def print_machine_info():
    host_name = socket.gethostname()
    ip_address = socket.gethostbyname(host_name)
    print ("Host name: %s" % host_name )
    print ("IP address: %s" % ip_address )

if __name__ == '__main__':
    print_machine_info()
```

In order to run this code, you can execute the following from the command line:

```
$ python 1_1_local_machine_info.py
```

On my machine, the following output is shown:

```
Host name: mymac.litdom.lit.ie
IP address: 172.16.101.48
```

This output will be different on your machine depending on the system's host configuration.

3. How it works...

The import socket statement imports one of Python's core networking libraries. Then, we use the two utility functions, gethostname()andgethostbyname(host_name). You can typehelp(socket.gethostname) to see the online help information from within the command line. Alternately, you can type the following address in your web browser at http://docs.python.org/3/library/socket.html. You can refer to the following command:

```
gethostname(...)
gethostname() -> string
Return the current host name.
gethostbyname(...)
gethostbyname(host) -> address
```

Return the IP address (a string of the form '255.255.255.255') for a host. The first function takes no parameter and returns the current or localhost name. The second function takes a single hostname parameter and returns its IP address.

Finished ?

- Upload your completed code to Moodle in the following format (NOTE: Moodle will be configured to accept two files as a submission)
  1. Upload the python source file (do not zip this file)
  2. Upload a full-screen screen grab of your finished work that includes the terminal output AND the python code visible in a text editor. Use png format and do not zip this file either.