Networking for Software Developers

# Assignment 1 – Threads and requests and argparse.

## Due:

To be done as a group. (See e-centennial for submission instructions)

### Maximum Points:

There is a total of 25 points for this assignment.

### Premise:

There is a speed improvement when you use threads to download multiple files at the same time as opposed to downloading the files sequentially.

### Description:

This exercise consists of four parts:

6 Marks

* Write a function to download a single file. This function will take two arguments: the url of the file (in this case an image from bing) and the pathname (the name of the downloaded file). You will use the request library to do this.

To indicate to the user that the program is alive, you should display the name of the resulting file on the console.

* Second is to write a function to invoke the above function to download a set of images sequentially. That implies that this function will send the appropriate arguments to the above function. The urls will be

7 Marks

* Third is to write a function to download the same set of images using threads. You will measure the time taken for both downloads to decide if there is any advantage of using threads.

10 Marks

* Write the logic to check if the script is being executed, if it is, it checks for the command line arguments and process them appropriately and then runs the program accordingly.

7 Mark

Please download your images to a separate folder so, you can delete the entire folder afterwards.

You must use the following image URLs:

urls = [

'https://th.bing.com/th/id/OIP.z-dkECmUFma29zYrb27JkwAAAA?w=264&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.MhwSzfXnBG1MpuuA6IFi-AAAAA?w=218&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.m8b7Y9-81Q4UMCBMaFkw2QAAAA?w=198&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.XN9D7tH47WNJ8h214YgqTwAAAA?w=220&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.cFvfW8dARmuVtR3zOxfTSAHaE9?w=274&h=183&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.wcEy7Ow-TaAohBCz6USqCAAAAA?w=265&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.9FWt0sWpi4UOee5o3WdI-QHaFj?w=224&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.JYXSCIpGskpiOxYTw1vuwgAAAA?w=252&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.QjWOHkojgYSz1LhaypSB-gAAAA?w=190&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.Wlfm\_lF4VWlYLiPNfbmbDwHaHa?w=181&h=181&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.ZquJ\_NwCCyWfvpAEeU-vngAAAA?w=142&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.C6q29lesR7-Ork5YKuI6LwAAAA?w=257&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.A7o1Bm-XNr9A\_4pLPCCujgAAAA?w=252&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.oSjt2rY3YUScDY7pw3b1WAHaFj?w=236&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.AroTG9KnmisPIhICyGjoDAHaFj?w=223&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.zSyHBN9\_rn\_O9XBkdPx-agAAAA?w=189&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.pGTxkbwreLj7l2ORZrtA8gAAAA?w=147&h=184&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.5SaLUh616MU7KDIP2\_0VCwAAAA?w=204&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.S-lrZd2TFhSEpI3VRQyKqQAAAA?w=173&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.sDmZWxXBrF329vZvDu2HrAAAAA?w=266&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.5umgRLykyWn-v\_5HmOS0NAHaE7?w=243&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.MPayRq2bYdhfVUj5O9BCnwAAAA?w=125&h=184&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.OeLv1q1dEfGkl1bRHfM5awHaFj?w=240&h=180&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.MksSZEmu5Cgly2HNvRp4NQAAAA?w=180&h=163&c=7&r=0&o=5&pid=1.7',

'https://th.bing.com/th/id/OIP.HCpPn-IRV8SVidBlRoBRUwHaE7?w=287&h=191&c=7&r=0&o=5&pid=1.7'

]

### Python frameworks/Modules:

* **requests**: This is an external framework that you would have need to run the labs. This would have been on your machine because it was used in previous labs.
* **time**: This is an internal module that is a part of python standard distribution and does not require any separate installation.
* **argparse**: This is an internal module that is a part of python standard distribution and does not require any separate installation.
* **threading**: This is also an internal module that is a part of python standard distribution and does not require any separate installation.
* **NO OTHER LIBRARY IS PERMITTED**!!!. Unless you justify to me in your code.

You may use the following pseudocode as a guide:

### Part A

1. Create a function to take a URL and a pathname that does the following.
   1. Use the request library to download the content (specified by the argument) as bytes.   
      *[see wk05\_a6\_download\_file.py]*
   2. Write the contents of the uri to the above file name (specified by the argument).   
      *[See wk05\_a6\_download\_file.py]*

If there is a problem with one of the urls in the list, then you may substitute it with one of your own.

* 1. Print a suitable message.

### Part B

1. Create a function that does not require any argument and does the following:
   1. Start timer.  
      *[import the time module  
      use time.perf\_counter() to get the current time.]*
   2. Call the function in Part A with appropriate arguments (first the supplied url list and the second one that you will have to generate) to sequentially each url. (Invoke the method after the previous download is completed).
   3. End timer.   
      *[again use time.perf\_counter() to get the current time.]*
   4. Print elapse times.   
      *[subtract (a) from (c).]*

With my present internet connection, I got a time of 2359 milli seconds.

### Part C

1. Create a function that does not require an argument and does the following:
   1. Start timer.
   2. In a threaded fashion, call the function in Part A with appropriate arguments.  
      *[see wk03\_d2\_threading.py.]  
      If your system complains about replacing the files in Part 2, you will need to remove/delete the files before running this part*
   3. End timer when all the threads have completed their downloads.
   4. Print elapse times.

With my present internet connection, I got a time of 232 milli seconds.

See the hints for Part B.

1 Mark

### Part D

1. If the script is running as an executable, it does the following:
   1. Creates an ArgumentParser and adds two arguments
      1. A mandatory argument that may be either serial or threaded
      2. An optional argument that allows the user to specify a data folder to download the images to
   2. If the mandatory argument is supplied, then invoke the appropriate method (either Part B or Part C).

### Deductions

* You will be penalized for poor quality of code.
* Your output must be clear and understandable to the user.

### Bonus

1. Any improvements on the above pseudocode. You need to highlight your improvement to the professor.

#### Submission

1. Your code file will be named «your\_group\_number»\_threads.py. e.g. group1\_threads.py.
2. The file must have a header stating all the members of the group who participated in the project.
3. Must be uploaded to course dropbox.
4. This is due by the end of next week.
5. Do not include the images in your submission.