Programming Exercise 8

Concurrent Programming

Background

In this assignment, you will be downloading GIFs for flags from all the countries in the world from a public website. The files appear in the following folder:

https://www.sciencekids.co.nz/images/pictures/flags96/

The names of the files have the following form:

```
    country-name.jpg
```

where **country-name** stands for the country name. For example, the JPG for the United States is *United_States.jpg*, and the full URL is *https://www.sciencekids.co.nz/images/pictures/flags96/United_States.jpg*. The country names are in the file *flags.txt* in the Canvas folder for this assignment.

Requirements

You will write three similar scripts that do the following:

- Download all flag files into a local directory named flags
- Report the number of bytes downloaded
- Report the execution time of the script (using **time.perf_counter()**)
- Report the CPU time of the script (using **time.process_time()**)

The only difference between the scripts is that for two of them, you will be using concurrent programming. The three versions will use the following respective schemes for downloading:

1. Download sequentially (no concurrency)
2. Download concurrently using futures with processes
3. Download concurrently using futures with threads

At the end of the third script, after everything is working, include code to send a mime email of the US flag (with its file name) to chuck.allison@gmail.com. The text portion of the message should appear before the attachment, and should say “Here you go, Professor Allison!”. Be sure to replace your password with asterisks before submitting your code! If you can’t be identified by your email, include your name in the body text as well. Turn in your three scripts and three output sets.

Implementation Notes

You may have to temporarily turn off your email security if you use secure servers like gmail.